Teensy BalanceBot Mk1

Курсов проект по
Вградени и автономни
системи

Никола Тотев

Софтуерно Инженерство
3 Курс

ФН: 62271

Въведение

Документацията, която следва предоставя на читателя подробна информация за проекта „Teensy BalanceBot Mk1”

Целта на проекта е да разработи балансиращ се робот от нулата. Това означава да се мине през пълния процес на създаване на продукт/проект. Всичко от анализ на системата и разработка й, до провеждане на тестове за проверка на функционалност и създаване на подробна документация на всички стъпки.

Анализ на системата

Тази секция покрива желаната функционалност на системата, как може да се моделира робота, какви решение съществуват и как те се сравняват с проекта, който се разработва.

Желана функционалност

Общ преглед

Това е сравнително малък проект, който разполага с ограничено време за разработка и тестване, затова желаната функционалност е ограничена. Характеристиките, които са избрани са достатъчни за създаването на MVP(Minimum Viable Product) - МЖП (Минимално Жизнеспособен Продукт), който може да служи за добра начална точка за бъдещи проект, които са базирани на балансиращ се робот. Бъдещи възможности за развитие и подобрения се обсъждат по-късно в документацията.

Възможни приложения

Въпреки малкия размер на Teensy BalanceBot Mk1, той има няколко възможни приложения

- Първото приложение е в образованието. Робота достатъчно прост, за да може да служи като въведение в роботиката за ученици и студенти, но има много възможности за бъдещо развитие.

- Второто приложение е в изследователската дейност. Роботите стават все по-обичайна гледка в ежедневието ни и различни ситуации изискват роботи с различни характеристики. BalanceBot Mk1 предоставя добра платформа върху която изследователите могат да надграждат.

Функционалност

- **Контрол на ъгъла** – Контрола на ъгъла е най-важната функционалност на Teensy BalanceBot Mk1. Контролирането на ъгъла не означава роботът да седи само в изправено положение. Помислете за случая, когато роботът трябва да се придвижи от точка А до точка Б. При преместването, поради естеството на робота, той трябва да се наклони или напред или назад. Това изисква повече внимание да се обърне по време на разработката, защото ъгъла на наклон не е константа и зависи от скоростта на робота.

- **Задържане на позиция** -Тази функционалност не е критична за робота, но е важна част от основната функционалност на робота. Възможността да се държи дадена позиция позволява за по-лесно настройване на контрол на ъгъла и тестване на бъдеща функционалност. Както беше споменато преди, тази платформа намира приложение в образованието и изследователската дейност. В тези случаи мястото за работа е ограничено и не е желателно робота да се измества по време на тестването.

Модел на системата

Създаването на модел на системата е критична стъпка от процеса на разработката. Той позволява потенциални проблеми да се открият на по-ранен етап, преди време и усилия да се вкарани в разработката на физически обект. Моделът също позволява да се направят симулации на системата. Тези симулации предоставят ценна информация, за това как системата се държи в реалния свят.

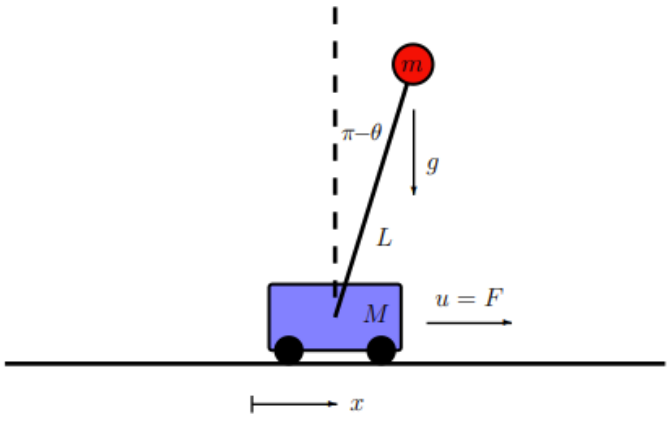
В случая на BalanceBot Mk1, симулациите дават възможност да се симулира как робота се контролира и как той реагира на даден физически импулс.

Математически модел

Създаването на математически модел е първата стъпка в създаването на цялостен модел на системата. Това е така, защото движението на всеки обект може да бъде описано със система от уравнения. Ако тези уравнения са достатъчно точни, симулацията също точно ще представи истинската системата.

Модел с обърнато махало

Преди уравненията, които описват системата е важно системата да се разгледа в по-опростен вариант. В този случай е добре да се разгледа схемата на свободно тяло на фиг. 1. Вижда се, че системата може да се представи, като обърнато махало поставено на количка.



Фиг. 1 Схема на свободно тяло (Steven L. Brunton, 2017)

Колелата на робота могат да се представят като количката, а останалата част от тялото като махалото. Този модел не е свръхточен, но е достатъчно добър за целите на този проект.

Ако се изисква по-голяма точност, повече време трябва да се отдели на математическия модел.

Уравнения на движението

Уравненията на движението описват поведението на физическата система от гледна точка на нейното движение като функция на времето. (R.G. Lerner, 1991). Извеждането на тези уравнения от схемата на свободно тяло е трудоемък процес и извън обхвата на този проект, затова се използват уравнения, които вече са изведени

$$\dot{x} = v \tag{8.67a}$$

$$\dot{v} = \frac{-m^2L^2g\cos(\theta)\sin(\theta) + mL^2(mL\omega^2\sin(\theta) - \delta v) + mL^2u}{mL^2(M + m(1 - \cos(\theta)^2))} \tag{8.67b}$$

$$\dot{\theta} = \omega \tag{8.67c}$$

$$\dot{\omega} = \frac{(m + M)mgL\sin(\theta) - mL\cos(\theta)(mL\omega^2\sin(\theta) - \delta v) + mL\cos(\theta)u}{mL^2(M + m(1 - \cos(\theta)^2))} \tag{8.67d}$$

Фиг. 2 – Уравнения на движението (Steven L. Brunton, 2017, p. 352)

Тези нелинейни уравнения на движението могат да се използват за моделиране на системата и създаването на симулация. Повече детайли за създаването на симулацията могат да бъдат намерени в следващите секции.

Възможни контролери

Когато става въпрос за контролери, има няколко възможности. Тази секция се фокусира върху сравнението на трите възможности, анализирайки плюсовете и минусите както и колко лесно е да се реализират.

Контролируемост на системата

Това е типичен проблем в сферата на теорията за контрол. Преди да се разработи система, която се нуждае от някакъв вид контрол, важно е да се провери дали изобщо е възможно тя да се контролира. Тази проверка може да се направи като се изчисли рангът на матрицата на контролируемостта използвайки MATLAB командата показана във фиг. 3.

В този случай рангът трябва да е 4 и при пускане на кода, той наистина извежда 4. Този тест потвърждава, че системата може да се контролира.

```
sys = ss(A, B, C, D);
ControlabilityMatrix = ctrb(sys);
rank(ControlabilityMatrix)
```

Фиг. 3 – ctrb() функция в MATLAB

Във фиг. 3 „sys” е представяна на системата в пространство от състояния и матриците A и B идват от уравненията на движението показани по-рано. Матриците C и D са показани на фиг. 4.

```
C = [1,1,1,1];
D = zeros(size(C,1), size(B,2));
```

Фиг. 4 - Матриците C и D

Детайли свързани с модела в пространството от състояния и математическата теория са извън обхваната на тази документация, но повече информация може да бъде намерена в книгата “Data Driven Science & Engineering” *(Steven L. Brunton, 2017, p. 323)*.

Оптимален пълен контрол на състоянието – (ЛКР)/(LQR)

Оптимален пълен контрол на състоянието се осъществява с Линеен Квадратичен Регулатор (ЛКР) - (LQR) Linear Quadratic Regulator. С този метод собствените стойности на матрицата на системата със затворен кръг (A-BK) се манипулират с избор на закон за контрол на пълното състояние u=-Kx. *(Steven L. Brunton, 2017, p. 343)*.

За да се осъществи този метод, трябва да може да се измери пълното състояние на системата. Това изисква добри сензори, които имат малко шум в измерванията. Тъй като се използват компоненти (сензори) от потребителски клас този метод на контрол не е подходящ. Компонентите и сензорите се разглеждат по-късно в този документ.

Оптимален контрол с пълна оценка на състоянието (LQG)

Пълна оценка на състоянието се осъществява използвайки филтъра на Kalman (Калман). Пълната оценка на състоянието изчислена с помощта на този филтър се използва съвместно със закона от ЛКР и крайния резултат с оптимална сензорно базирана обратна връзка. (Steven L. Brunton, 2017, p. 350). С тази обратна връзка се осъществява контрол на системата. Този метод изиска повече анализ за да се установи колко е наблюдаема системата. Възможно е пълното състояние на системата да не може да бъде оценено при липсата на различни измервания. Този метод също зависи на добри сензорни данни и затова първата итерация на този проект не имплементира този начин на контрол.

ПИД (PID) Контролер

ПИД контролера е метод на контрол, който използва обратна връзка да контролира системи, които се нуждаят постоянно модулиран контрол. Този метод е широко разпространен в индустриални системи за контрол както и в други приложения като роботиката. Той е най-простия за имплементиране от трита метода разгледани в тази документация. Основното предимство на ПИД контролерите е, че по-лесно се имплементират на микроконтролер.

Подобен контролер не изисква сензори с голяма точност и дори и сензори с повече шум могат да се използват.

Поради по-лесната имплементация и олекотените изисквания за сензорите, Teensy BalanceBot Mk1 използва ПИД контролер.

ПИД контролера както и целия модел на системата са направени в Simulink, защото има всичките нужни инструменти за създаването на модела използвайки вече наличните 3Д модели. Повече информация за избора на софтуера има в следващата секция.

Софтуер за моделиране

Тази секция разглежда използвания софтуер за създаване на модел на системата. Това включва CAD софтуер, както и програми за моделиране на поведението и контрол на системата.

Софтуера за моделиране е важна част от разработката на която и да е система. Предоставя евтина начин да се създаде виртуална версия на системата. Този виртуален близнак, помага за бъдещето създаване на системата, например 3Д моделите разработени в CAD могат да се използват при 3Д принтирането. Същите 3Д модели могат да се вкарат в софтуер като MATLAB и Simulink за да се симулира поведението на системата.

Autodesk AutoCAD

CAD софтуера е най-важния от всички за този проект. Това е така, защото фазата на производство, така и фазата на моделиране в MATLAB зависят от 3Д моделите на системата.

Първата причина да се избера Autodesk AutoCAD беше, че проектантът, единствения човек работещ по този проект има опит с тази програма. Втората е, че тя има всичката функционалност, която проекта изисква.

MATLAB & Simulink

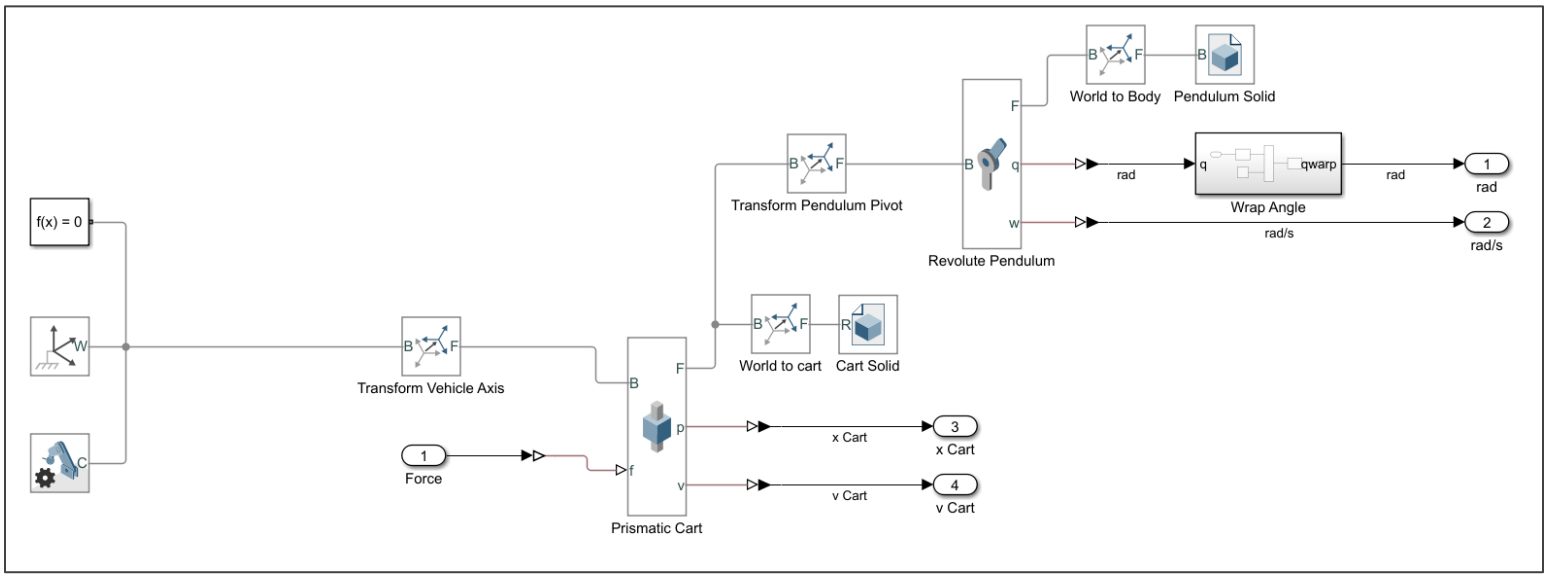
За създаването на модела на системата и да се моделира динамиката и контрола се използва MATLAB, Simulink и Simscape. Други продукти имат подобна функционалност, например Mathematica, но MATLAB има повече функционалност, специално направена за моделиране на системи.

MATLAB Модел и Симулация

Тази секция обсъжда MATLAB модела създаден с помощта на Simulink, резултатите от него и разработения ПИД контролер.

Simulink Модел

Simulink модела се раздела на две части. Първата част е модела на робота, а втората ПИД контролерите. Първата част е показана на фиг. 5, а втората на фиг. 6. По-големи версии на схемите могат да бъде намерени на Github (Nikola, 2021) страницата на проекта.



Фиг. 5 – Модел на робота (no control)

Модел на контролера

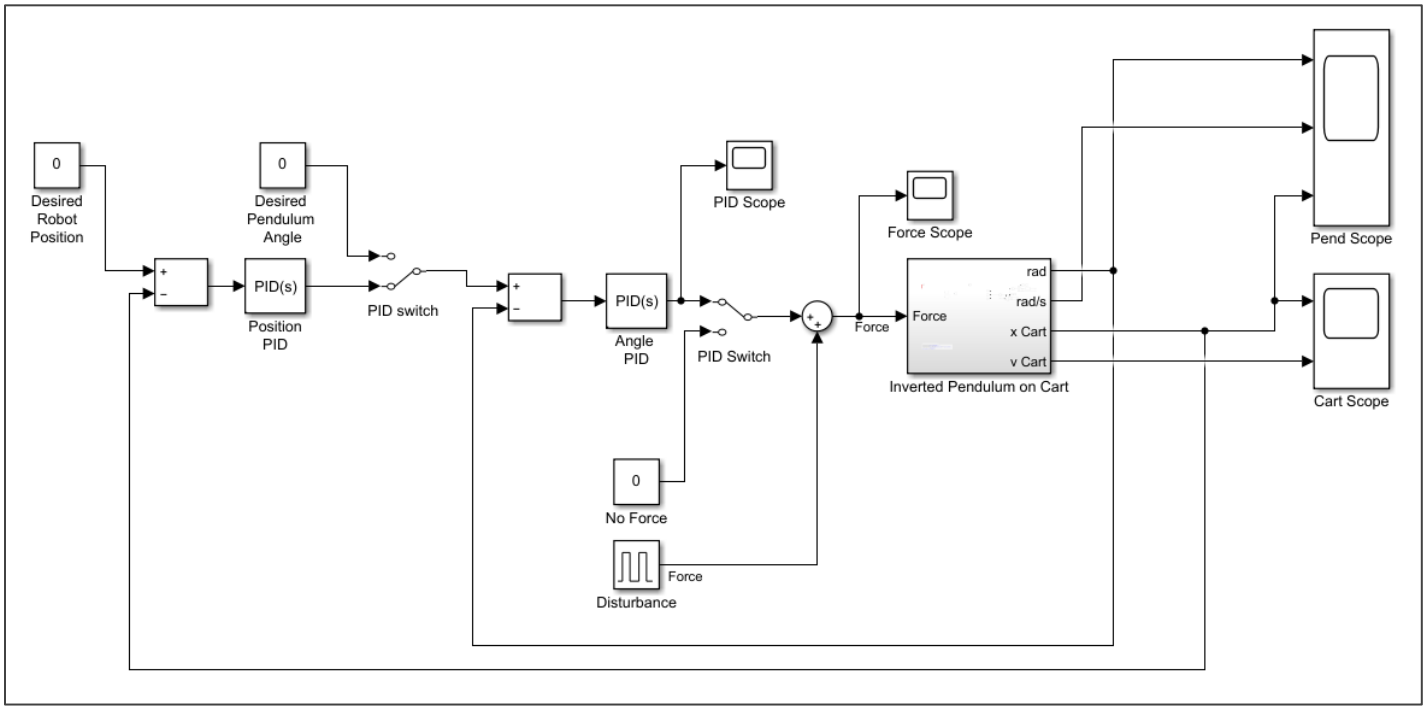
Тази секция разглежда модела на контролера какви входни данни приема, как е настроен и какви входни данни се приемат от модела на робота.

Модел на ПИД контролера

Както се вижда на фиг. 6, има 2 ПИД контролера. Вътрешния цикъл е за контрол на ъгъла, а външния е за задържане на позиция. И двата контролера използват вградения ПИД блок в Simulink.

Входните данни за контрола на ъгъла са съставени от две части – желания ъгъл и измерения ъгъл. Измерения ъгъл идва от революционната става на системата. Желания ъгъл може да се избере ръчно да е 0 или изхода на контролера за позицията.

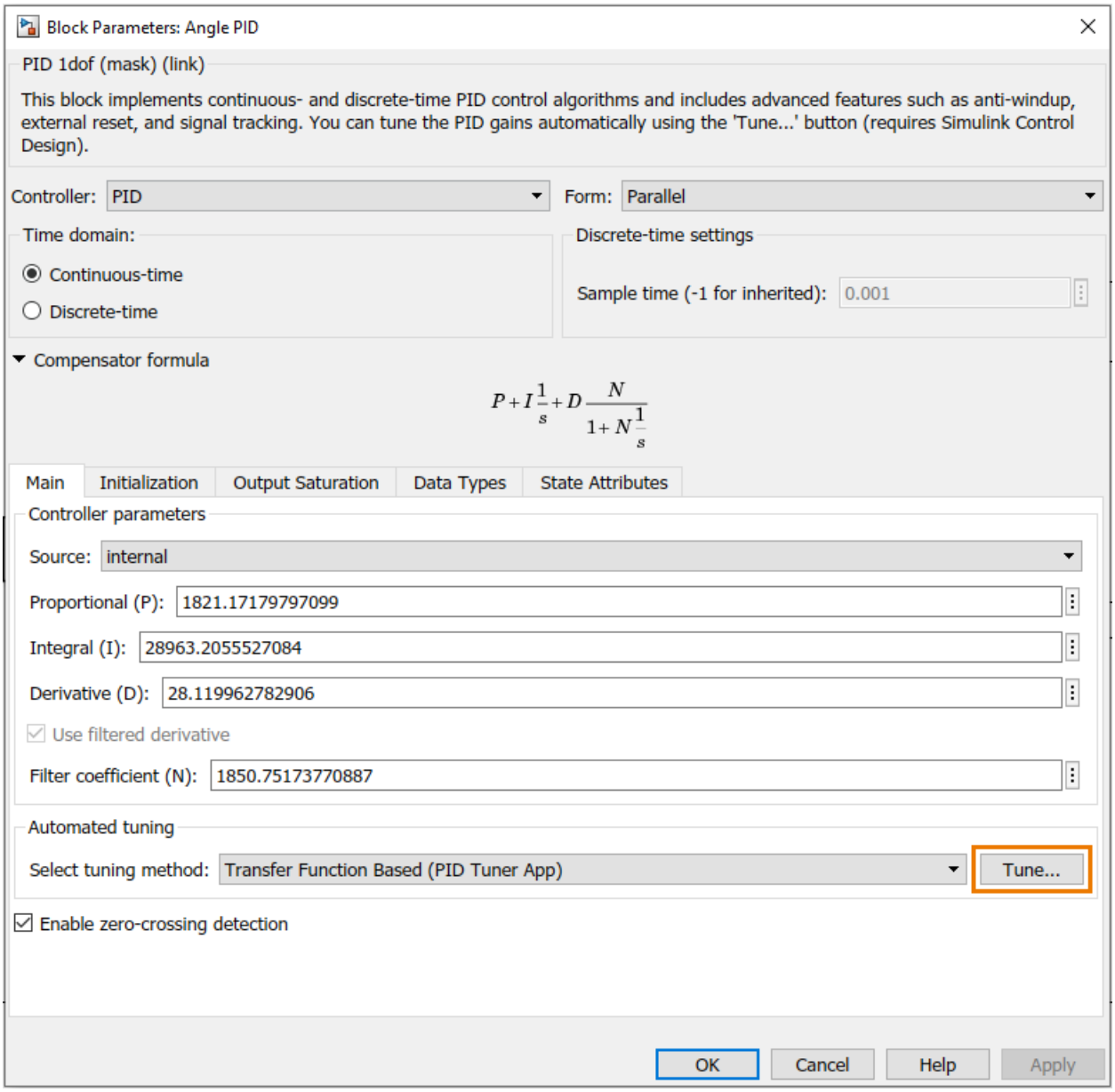
Контрола за позицията приема желаната позиция, която в този случай е константен блок със стойност 0 и измерената позиция на робота, която се взима от призматичната става на модела. Изхода, както беше споменато преди се подава като желан ъгъл на контролера за ъгъла



Фиг. 6 – ПИД контролери

Автоматична настройка на ПИД в MATLAB

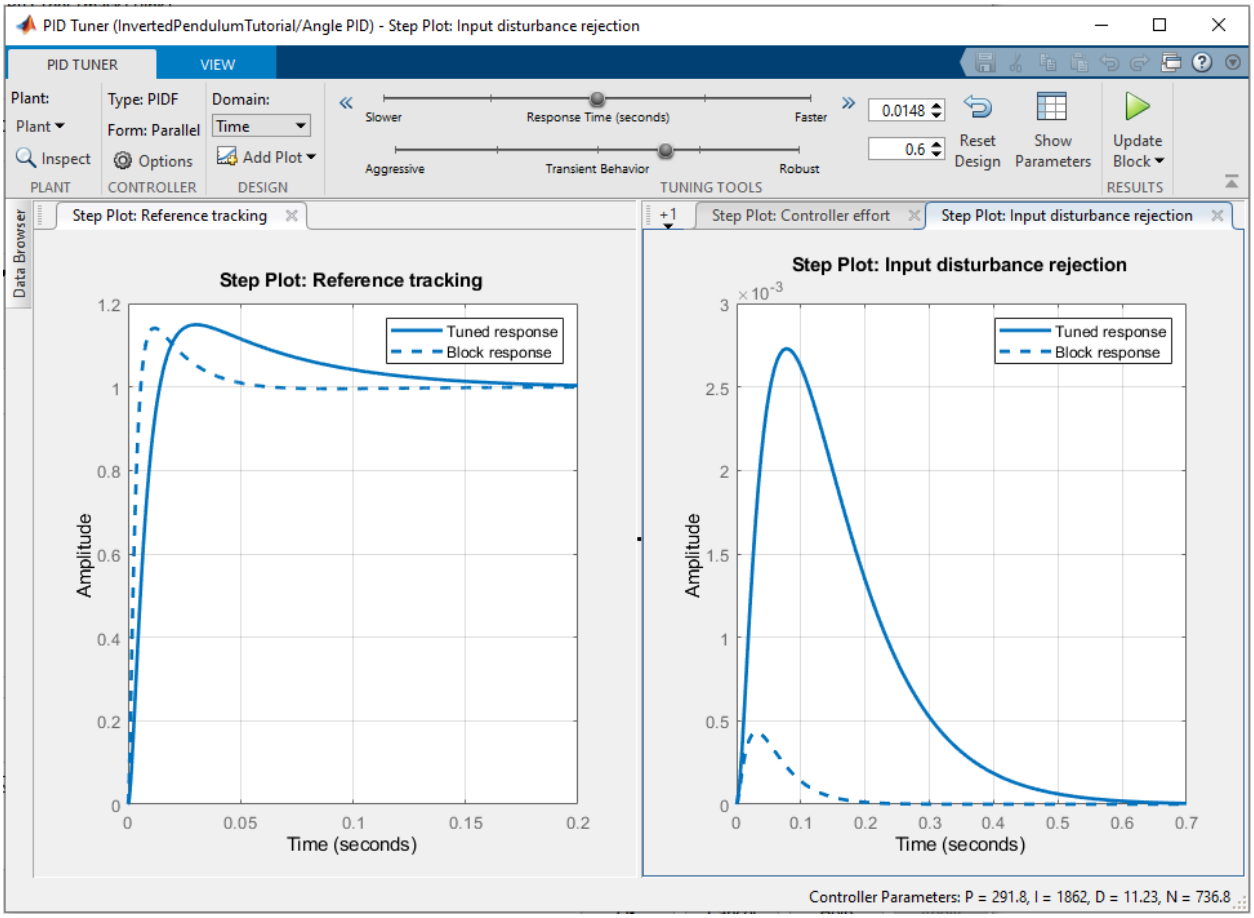
Ако болка на ПИД контролера се натисне два пъти, прозорец със свойствата на контролера се показва.



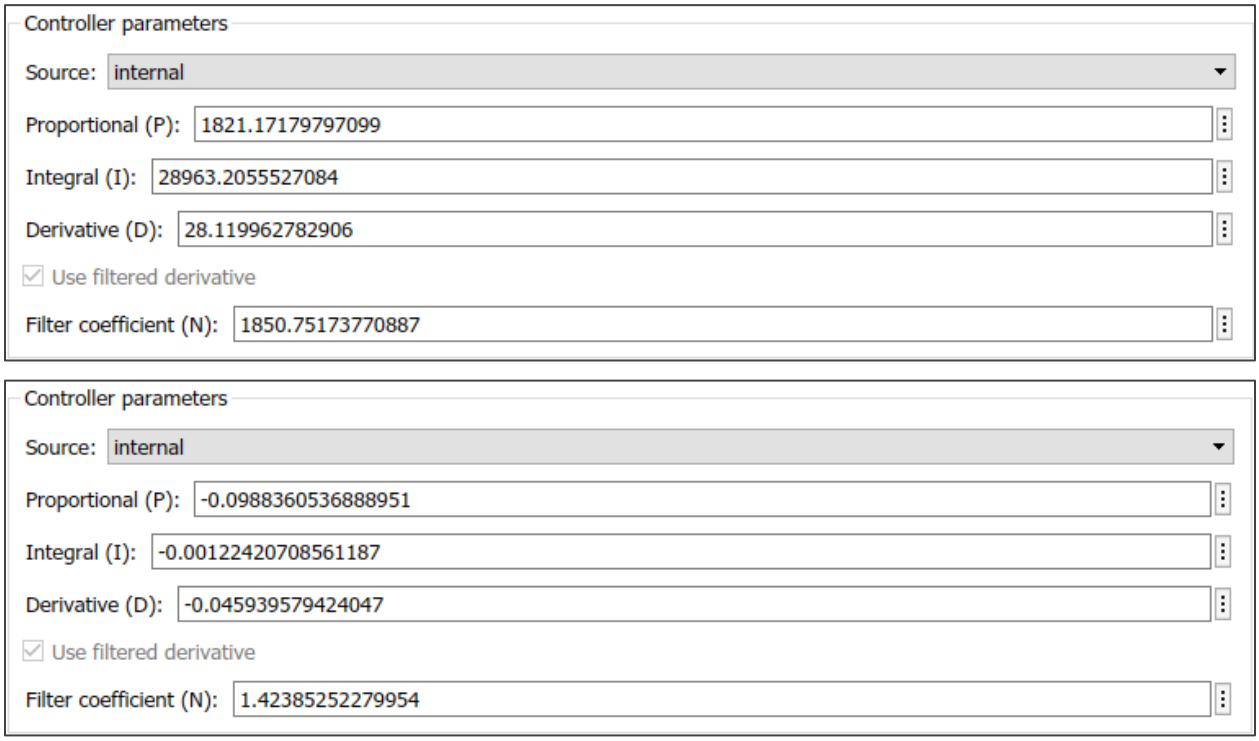
Фиг. 7 – Свойства на ПИД контролера

В долния десен ъгъл има бутон за настройване (tune button). Така автоматично се настройва ПИД контролера. Възможно е да се направи ръчна настройка, но е трудоемък процес и резултатите не са толкова точни.

След натискане на “Tune” бутона, нов прозорец се отваря (фиг. 8). От този прозорец различни параметри (колко бързо или бавно да реагира системата и колко да е стабилна) могат да се променят за да се промени поведението на системата. След тестване на различни стойности, на фиг. 9 могат да се видят крайните стойности, които се използват съответно за контролера за позицията и ъгъла.



Фиг. 8 – Прозорец за настройване



Фиг. 9 – ПИД за ъгъла (горе) и ПИД за позицията(долу)

Симулация

Симулацията се изпълнява в MATLAB и се визуализира като 3Д модел на робота. Преди симулирането на контролиран робот, се симулира без контрол за да се провери дали се държи по очакван начин. След като се установи, че поведението е правилно се включват ПИД контролерите и симулацията отново се пуска. Блок за смущения се добавя за да се симулира побутване на робота.

Резултатите от симулацията се записват като .mp4 файл.

Резултати от симулацията

Резултатите от симулацията могат да се видят в YouTube на следния линк:

https://youtu.be/9-L7qeeoDtc

или в Github страницата на проекта.

https://github.com/NikolaTotev/Teensy-Balance-Bot-Mk_1

Резултатите от симулацията показват, че без контрол, роботът пада и, че когато се активира ПИД контролера, роботът успява да се задържи във вертикална позиция и да задържа нулева позиция, а смущенията се компенсират.

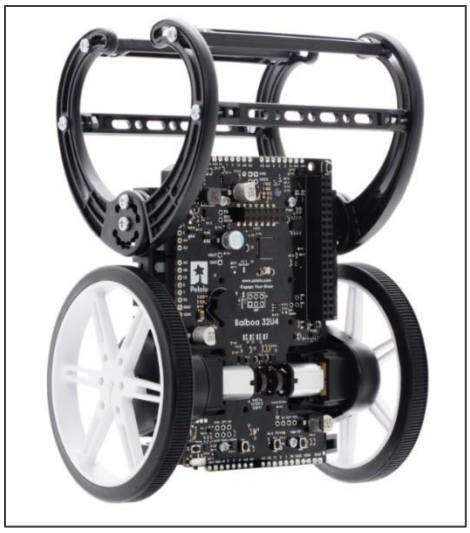
Сравнение със съществуващи решения

Тази секция анализира съществуващите решения, които са на пазара и имат подобна функционалност. Това сравнение ще вземе предвид както функционалността, така и възможните приложения, защото има много продукти, които имат подобна функционалност, но са предназначени за индустрията, имат много по-големи размери или и двете.

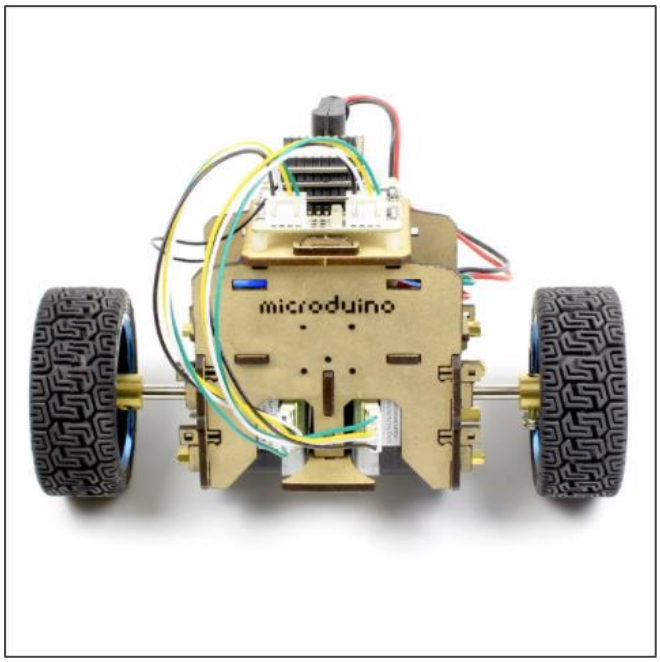
Съществуващи решения

Тъй като има ограничен брой продукти, които имат подобна функционалност и подобни приложения, само два съществуващи продукта ще бъдат сравнени.*)*

- **Balboa 32U4 балансиращ се робот от Pololu –** Това е комплект, който може да се сглоби, като колелата и моторите трябва да се закупят допълнително. До края на сравнението, този продукт накратко ще бъде наричан Balboa. *(Снимка от* (Pololu, 2021)*)*



- **Microduino балансиращ се робот от Microduino**. Това също е комплект, който се сглобява и идва с всичко необходимо. До края на сравнението, този продукт накратко ще бъде наричан Microduino. *(Снимка от* (Robotshop, 2021)*)*



Характеристики

Тази секция разглежда характеристиките на съществъващите решения и след това ги сравнява с Teensy BalanceBot Mk1 на база на списък от критерии които ще бъдат описани следващата подраздел.

Balboa

- Размер - 118 × 112 × 80 mm
- Тегло - 200g
- Микроконтролен - Arduino-compatible ATmega32U4
- Състои се само от една плакта.
- Вграден инерционен модул
- Вградени драйвери за мотори
- Вградени квадратични енкодери
- Интерфейс за Raspberry PI
- Използва четкови мотори
- 6 AA батерии

Microduino

- Размер - 19.5x8.5x11cm
- Тегло – 650 g
- Контролира се с Bluetooth
- Микроконтролер - Atmel ATmega1284P/ATmega644PA
- Радио модул: Microduino-nRF24
- Използва стъпкови мотори
- Две 3.7V Li-ion батерии.

Критерии за сравнение

От предходните списъци, двата екземпляра имат подобни характеристики, но има и разлики. За да се създаде равностойно сравнение между двата продукта и робота, който се разработва в рамките на този проект, ще се използват следните критерии:

- Модел на микроконтролера (MCU)
- Инерционен модул (IMU) и модел на модула (IMU Model)
- Тип използвани мотори (Motors)
- Тип драйвер за мотори (Motor Driver)
- Енкодери (Has encoders)
- Вид батерия (Battery Type)

Продукт	MCU Model	IMU & IMU Model	Motors	Motor Driver	Has Encoders	Battery Type
Balboa	ATmega32U4	Да - Неизвестен модел	Четков - 30:1 Micro Metal Gearmotor HPCB 6V	Вграден – Неизвестен модел	Има	AA – Презар. се
Microduino	ATmega1284P ATmega644P	Да – Неизвестен модел	Стъпков – Незнаен NEMA Модел	Вграден – Неизвестен модел	Няма	3.7V – Li-On
Teensy BalanceBot Mk1	Teensy 4.1 - ARM Cortex-M7	Да – Pololu LSM6DS33	Четков – Обикновен с предавки (неизвестна марка)	Външен – Cytron MDD3A	Има	3.7V – Li-On

Инструменти за софтуерна разработка

Тази секция накратко ще разгледа избора на инструменти за разработка на софтуера, тъй като е важна част от разработката.

Предимно се използва Visual Studio с приставката Visual Micro. Тази програма беше избрана пред други като Visual Studio Code с приставката Platform IO или Arduino IDE, защото предлага по-стабилна среда за разработка, както и повече функционалност, която улеснява процеса на разработка.

Разработка

Този раздел описва в детайли целия процес на разработката. Разделен е на две части – хардуер и софтуер. Всеки подраздел съдържа информация за детайлите взети предвид при разработката, всичко от избора на микроконтролер, сензори, актуатори до избора на материали, информация за сглобяване и разработка на софтуера

Хардуер

Тази секция разглежда процеса на разработка на хардуера на BalanceBot Mk1.

Микроконтролер

Избрания микроконтролер за Teensy BalanceBot Mk1, както името подсказва е Teensy 4.1. Този микроконтролер беше избран за проекта, заради следните характеристики:

- Има малък размери не изисква много място за инсталация, но въпреки малкия размер има много пинове и интерфейси, които могат да се използват за бъдещи допълнения, като светлини, екрани, актуатори или сензори.
- 32-битов процесор – Процесорът на Teensy 4.1 е 32-битов ARM Coretx M7 и работи на честота от 600Mhz и има много функционалност, най-важната от които е поддръжката на операции с плаваща запетая.
- Добра поддръжка – компанията, която произвежда Teensy 4.1 – PJRC има много добра поддръжка за продуктите си и имат активен форум, който помага при възникване на проблеми

Сензори

Тази секция обсъжда какви сензори се използват и защо.

Общ преглед и съображения

Тъй като BalanceBot Mk1 изисква само основна функционалност, само два сензора са необходими – инерционен модул (IMU) и енкодери за колелата. Поради ограничените ресурси, използваните сензори са от потребителски клас. Сравнително евтини са, но предоставят нужната функционалност.

IMU – Жироскоп и акселерометър

Инерционния модул, който се използва е LSMDS33 от компанията Pololu. Този инерционен модул има жироскоп и акселерометър, това означава, че с помощта на комплементарен филтър може да се осъществи sensor fusion.

Модула поддържа протоколите I2C и SPI. За този проект се използва I2C протоколът. Сензора има измерена стойност от 16 бита за всяка ос от жироскопа и акселерометъра. Предоставя допра точност за цената си и има малък размер, което позволява лесно да се монтира.

Енкодери

Teensy BalanceBot Mk1 използва комплект магнитни енкодери от Pololu. Предназначени са да се използват с мотори, които имат удължена ос, но те бяха единствените енкодери достъпни по време на разработката. За да се използват правилно, специална част беше направена, която да държи платката на енкодера и адаптер беше направен да държи магнитите на оста на мотора. Повече информация относно дизайна и създаването на 3Д моделите може да се намери в раздела „CAD Модели“ Проблемите, които изникват заради начина на използване на енкодерите се разглеждат в раздела „Бъдещо развитие“

Актуатори

Актуаторите са от съществено значение за роботите и избора на правилния актуатор зависи от приложението. Тази секция обсъжда възможни актуатори, как се сравняват помежду си и защо един от тях е избран вместо другия.

Общ преглед и съображения

Тъй като BalanceBot Mk1 е мобилен робот, способността за точен контрол на въртенето на колелата е важна. Моторите трябва да имат и достатъчно мощност за да контролират робота. В случая на този робот, ако се използват четкови постояннотокови мотори, те трябва да имат предавки, защото малките мотори не са мощни. Друго нещо, което трябва да се вземе предвид е възможността за следене на завъртането на мотора в сравнение с зададената команда. Както беше споменато в предходни раздели, това се осъществява с магнитни енкодари. Това е още едно ограничение при избора на мотор.

Мотори

Когато става въпрос за мотори, както беше споменато преди, една възможност е да се използват постояннотокови мотори с предавки, друга възможност е да се използват стъпкови мотори. Този проект използва постояннотокови мотори, заради енкодерите, които са достъпни. Друга причина да се използват постояннотокови мотори е мощност, стъпковите мотори са по-точни, но в повечето случаи не идват с предавки. По време на разработката на проекта, достъпните стъпкови мотори бяха по-слаби от постояннотоковите. Проблемите и възможни решения, които идват от използването на постояннотокови мотори се разглеждат в раздела „Бъдещо развитие“

Драйвер за мотор

Драйвера за мотор, който се използва в BalanceBot Mk1 е Cyrtron MDD3A. Може да работи с 4-16V, 3A и може да управлява два постояннотокови мотора или един стъпков мотор. Приема PWM сигнал като вход за определяне на скоростта и логическо ниско/високо ниво, за определяне на посоката. Драйвера има и вграден регулатор на волтажа, който доставя 5V.

CAD Проектиране и производство

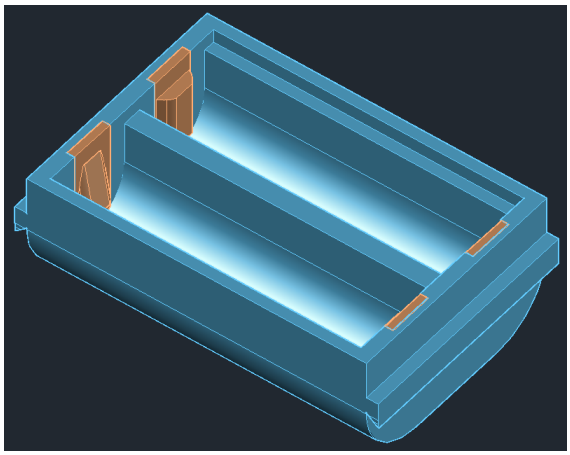
Този раздел разглежда структурата на робота, как е проектирана, основните части избора на материали и методи на производство.

CAD Дизайн

Тъй като BalanceBot Mk1 е съставен от малко на брой части, този подраздел ще ги разгледа по-подробно.

Отделение за батерията

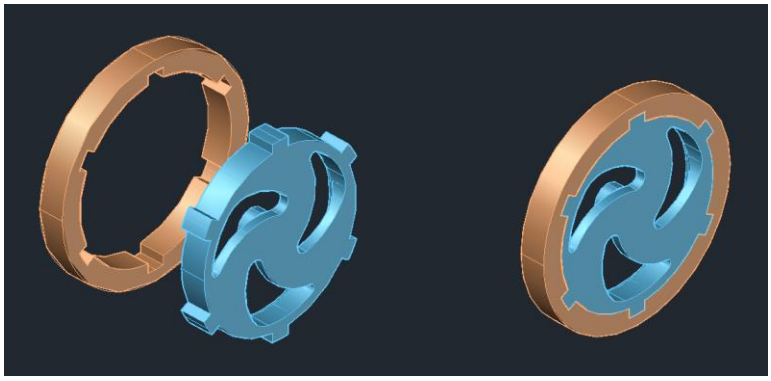
Поставката за батерии се намира в долната част на робота, тя държи 2 Panasonic NCR18650PF Li-On презареждащи се батерии. Батериите се държат от две пружини направени от TPU покрити с медно тиксо за провеждане не ток.



Фиг. 10 – Отделение за батерията

Колела

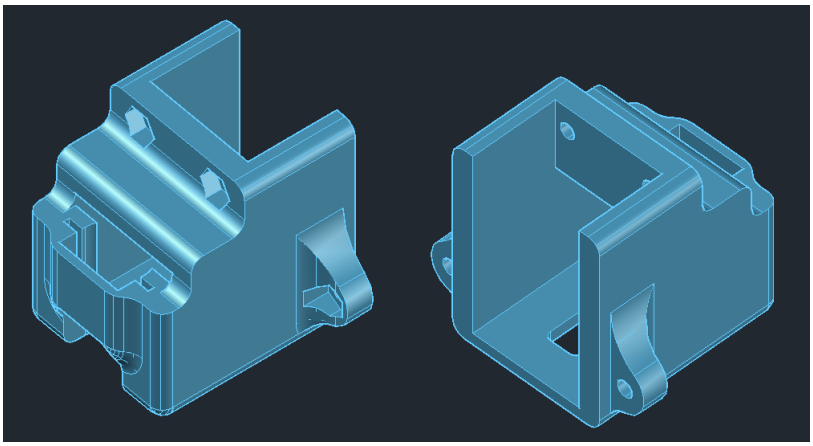
Колелата са с диаметър 100мм. Съставени са от две части – твърда пластмасова главина, която се закача с триене към мотора и мека гума направена от TPU, която се захваща за главината с правоъгълни прорези.Специални колела бяха проектирани, защото тези, които идват с моторите са прекалено малки. Благодарение на меките гуми и здрава главина, тези колела работят като стандартните.



Фиг. 11 – Колела и гуми

Поставка за енкодерите

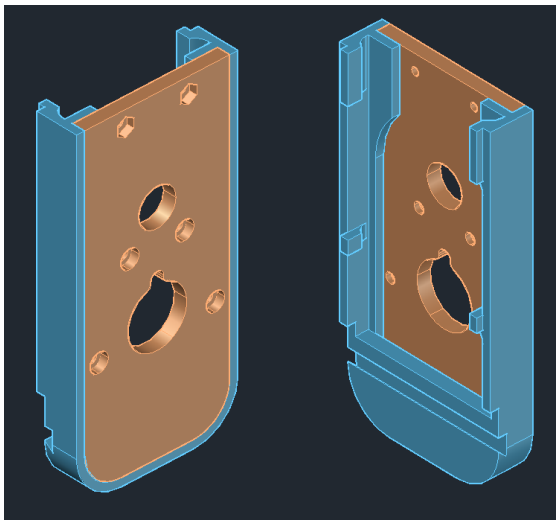
Поставките на енкодерите имат двойна функция, държат платката на енкодера на правилното разстояние от мотора и помагат за закрепването на моторите. Платката на енкодера е запоена за помощна платка и помощната платка влиза в жлеба на поставката. Има отвор за проводниците на енкодера. Няма нужда се допълнително закрепване на платките, защото триенето между частите ги държи достатъчно добре.



Фиг. 12 – Поставка за енкодер

Долни странични панели

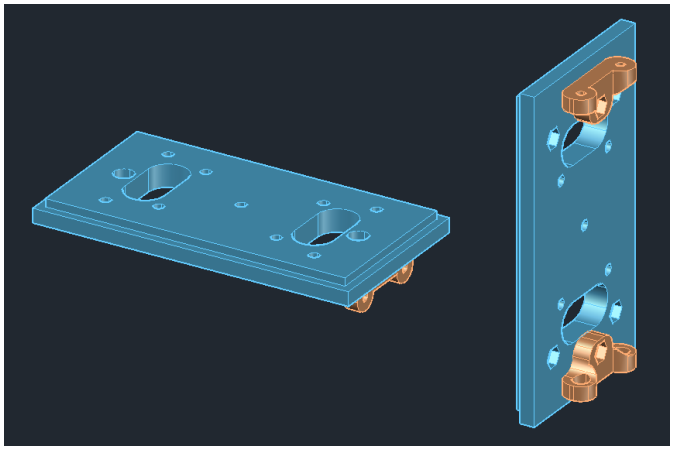
Долните странични панели държат моторите и поставката за енкодерите. Имат и трапецовидни релси, които служат за монтиране отделението за батерии. В горната част на панелите, скобите от средния панел се закачат.



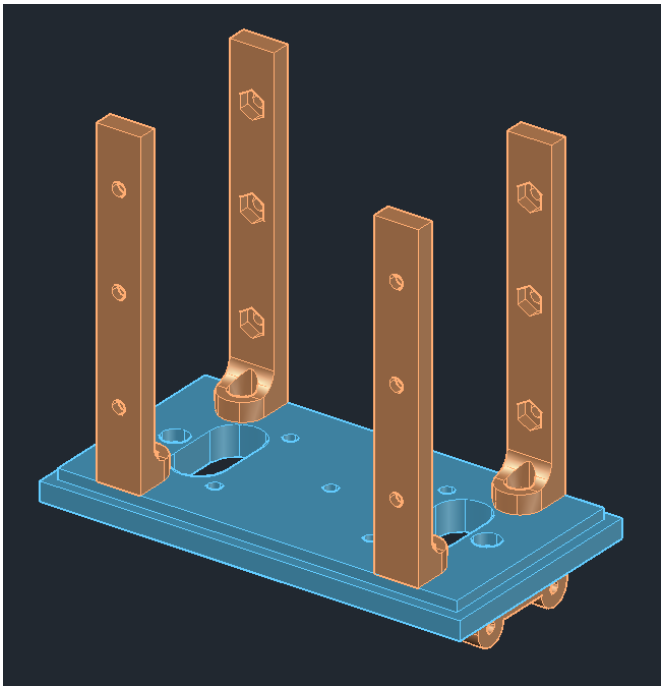
Фиг. 13 – Долни странични панели

Среден панел

Средния панел е интерфейса между долната и горната част на робота. Има отвори за кабели и плоска част за монтиране на breadboard, който държи инерционния модул (IMU). Има два различни типа дупки, които служат за монтиране на крепежни скоби. Един тип за скобите да долната част на робота и втори за скобите на горната част на робота.



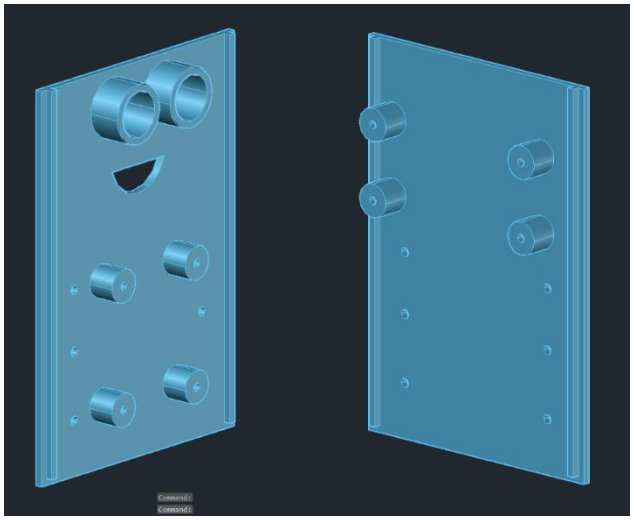
Фиг. 14 – Среден панел с долни скоби



Фиг. 15 – Вертикални скоби

Горни панели (преден и заден)

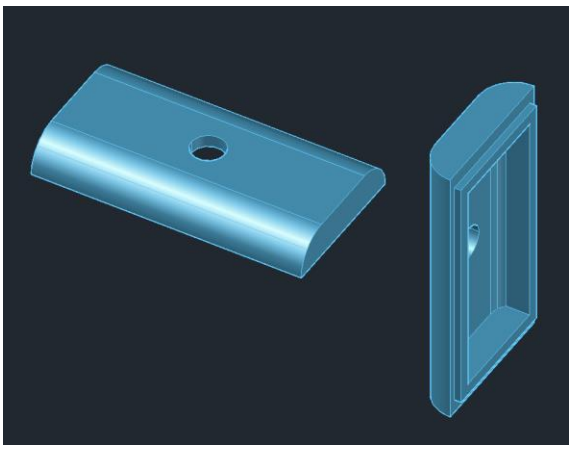
Горните панели създават отсека за електрониката, който държи микроконтролера и драйвера за мотора. Има дупки за закачана на платките с дистанционни болтове. Горните панели, както беше споменато преди се закачат с вертикални скоби за средния панел (има две скоби за всеки) и 6 болта се използват за закрепването на панелите към скобите.



Фиг. 16 – Преден и заден горен панел

Горен капак

Горния капак се използва за монтиране на главния ключ за захранването, както и да добавяне на допълнителна тежест в горната част на робота



Фиг. 17 – Горен капак

Метод на производство

Метода на използван за производството на BalanceBot Mk1 е 3Д принтиране. Използва се, защото позволява за бързо създаване на прототипи и тестване на промени. То също е евтин и достъпен начин за създаване на сложни части.

Избор на материали

Материалите използвани за частите на BalancBot Mk1 са PLA и TPU.

PLA беше избран, защото по-лесно за принтиране от ABS. PLA е по-слаб от ABS, но този робот няма части, които изпитват големи натоварвания.

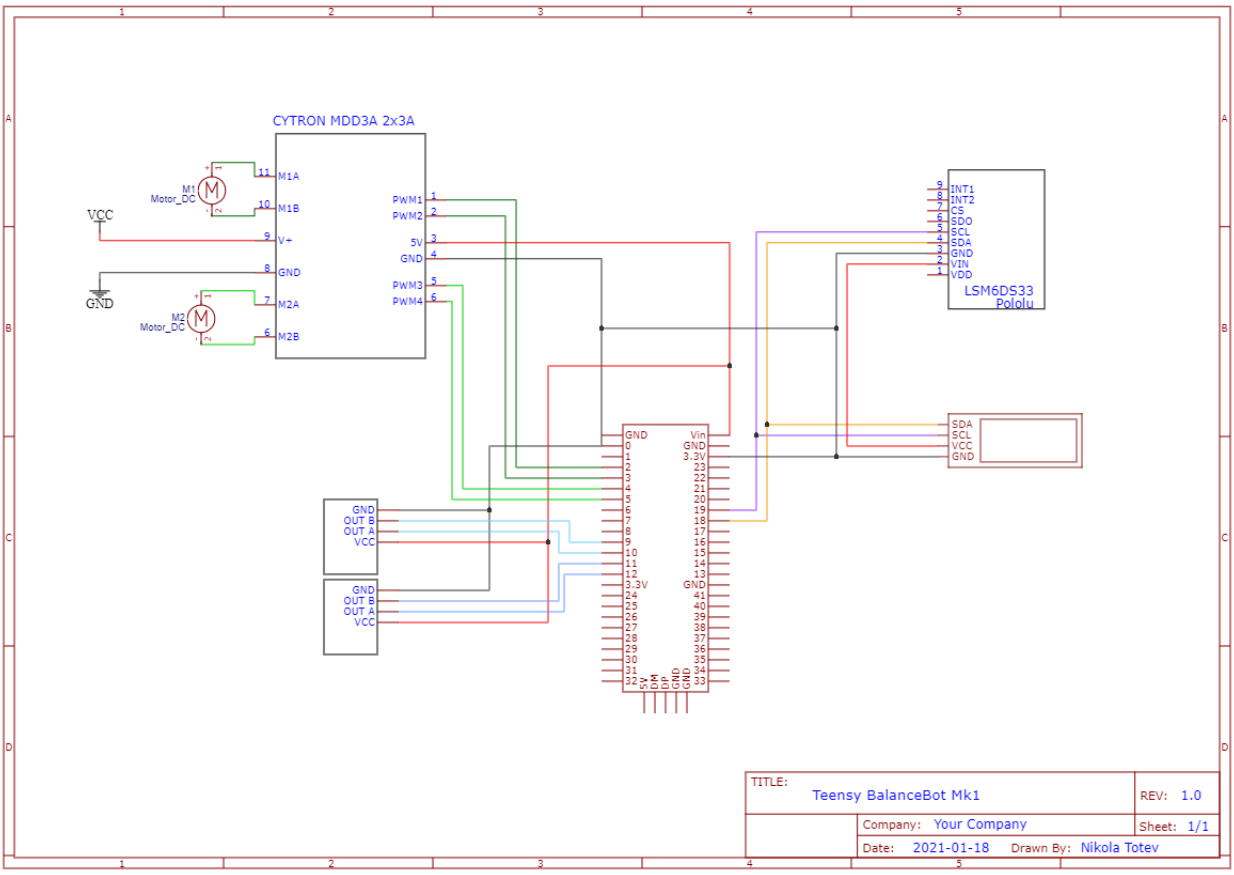
TPU с твърдост 70A на скалата на Shore се използва. TPU се използва предимно за гумите, защото те трябва да са меки да осигуряват сцепление. Този материал трудно се принтира, но с правилните настройки, резултатите са приемливи.

Електроника

Този раздел съдържа информация за електрониката на BalanceBot Mk1. Разглеждат се връзките между отделните компоненти и какви конектори и проводници за избрани

Диаграма на компонентите

Фиг. 18 показва диаграма на компонентите. По-голяма версия на схемата може да бъде намерена на Github (Nikola, 2021) страницата на проекта.



Фиг. 18 - System circuit diagram

Монтиране на микроконтролера

Микроконтролера има пинове, които се вкарват в рейка. За по-лесна инсталация и деинсталация, както и улеснено тестване се използват двойни редове от рейки. Рейките за запоени за малка платка с размери 3x7см. Женски JST конектори са запоени за платката, те служат за лесно закачане на останалите компоненти за микроконтролера.

Съединители и проводници

- **Съединители (конектори)**

Конекторите, които се използват в BalanceBot Mk1 за JST конектори с 2 и 4 пина, и Dupont конектори. Енкодерите и инерционния модул използват JST съединители, и мъжки JST към Dupont конектор се използва за връзката с драйвера на мотори.

JST конекторите осигуряват здрава връзка между компонентите, за разлика от Dupont съединителите, които лесно могат да се разделят. Единствената причина за използване на Dupont конектори е факта, че драйвер идва с такива.

Dupont конекторите също се използва за съединяването на breadboard върху който е монтиран инерционния модул. Проблемите, които възникват от този начин на инсталиране на IMU модула се разглеждат в раздела „Бъдещо развитие“

- **Проводници**

Използваните проводници са многожилни и са със силиконова или PVC изолация. Препоръчително е изолацията да е само силиконова, защото е по-гъвкава и по-лесно се работи с нея в тесни пространства, но кабели с PVC изолация трябваше да се използват, защото Dupont конекторите идват с такива кабели.

В процеса на разработка, както се очакваше, кабелите със силиконова изолация бяха по-издръжливи и по-лесни за работа.

Захранване

Батерии и поставка за батерии

Както беше споменато преди, отделението е направено от PLA и пружините за задържане на батериите са направени от TPU. Върху пружините е залепено медно тиксо и то е запоено серийно за. Отделението за батерии има мъжки JST конектор, който се закача за окабеляването закачено за главния ключ и драйвера за моторите.

Батериите, както беше споменато преди са 2 NCR18650PF 3.7V Li-On батерии с капацитет от 2700mAh

Изисквания към захранването

Мощността, необходима за нормална работа, е около 0.8-0.9A при максимално натоварване.

Всеки мотор използва до 350mA.

Микроконтролера изисква 100mA и се захранва от 5V, които се предоставят от вградения в драйвера регулатор на волтажа.

Сглобяване

Този раздел разглежда подробно детайлите относно сглобяването на робота. Избора на крепежни елементи и стратегии за сглобяване се обсъждат и сглобения робот е показан.

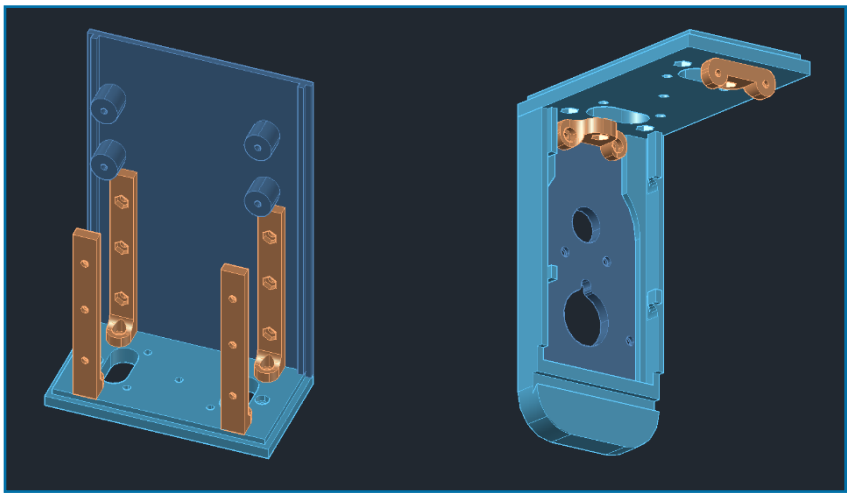
Избор на закрепващи елементи

Има много възможности за крепежни елементи. За размера на BalanceBot Mk1 и взимайки предвид толеранса на 3Д принтера, най-добрия избор са M3 болтове и гайки. Вдлъбнатина се създава в която влиза гайката и подобна се прави за главата на болта, за да не стърчи навън.

CA (cyranoacrylate) лепило се използва за залепяне на частите от страничния панел.

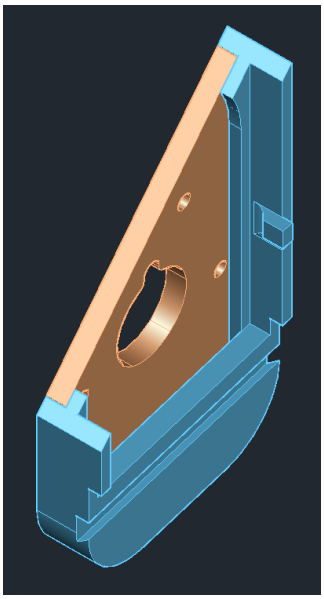
Стратегии за сглобяване

Както беше споменато в преди, използват се болтове и гайки. Основната стратегия е да се използва централна част (средния панел), към него се закачат крепежни скоби, а върху тях се закрепят горната и долната част. Това може да бъде видяно на фиг. 19.



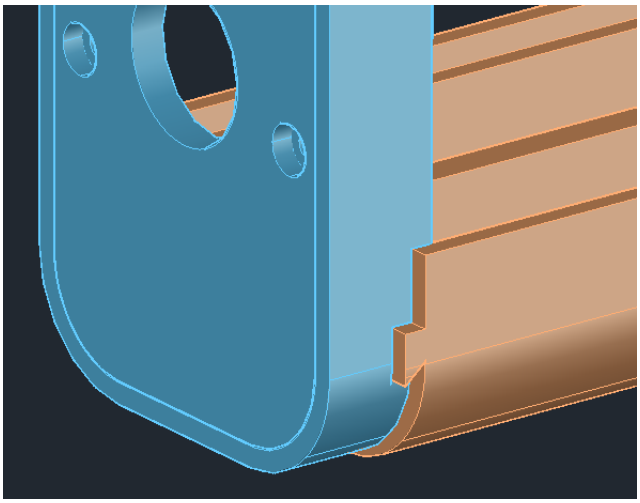
Фиг. 19 - Bracket assembly

Следващата стратегия може да бъде видяна на фиг. 20, малък ръб се прави на вътрешността на частта и върху не ляга външния панел и така се създава солидна плоскост. Когато се използват подобни ръбове, се добавя и CA лепило се по-голяма здравина.



Фиг. 20 - Notch assembly

Последната стратегия се вижда на фиг. 21. Трапецовидна релса се използва, когато нещо трябва временно да се монтира, но лесно да се маха. В този случай се използва за отделението на батериите, тъй като трябва да може да се премахва, когато е време за презареждането им.



Фиг. 21 - Dovetail assembly

Стъпки за сглобяване

Стъпка по стъпка инструкции могат да бъдат намерени на Github страницата на проект. Името на файла е AssemblyManual.pdf.

Сглобения робот

Фиг. 22 показва сглобения робот./

Фиг. 22 - Assembled robot

Софтуер

Този раздел разглежда подробно процеса на разработка на софтуера на BalanceBot Mk1.

Стратегия за разработка

Стратегията която се използва за разработка на софтуера е първо да се разработят отделните части и да се направи проверка дали работят правилно поотделно, след което да се съденият заедно. Различните компоненти се разглеждат в следващия раздел.

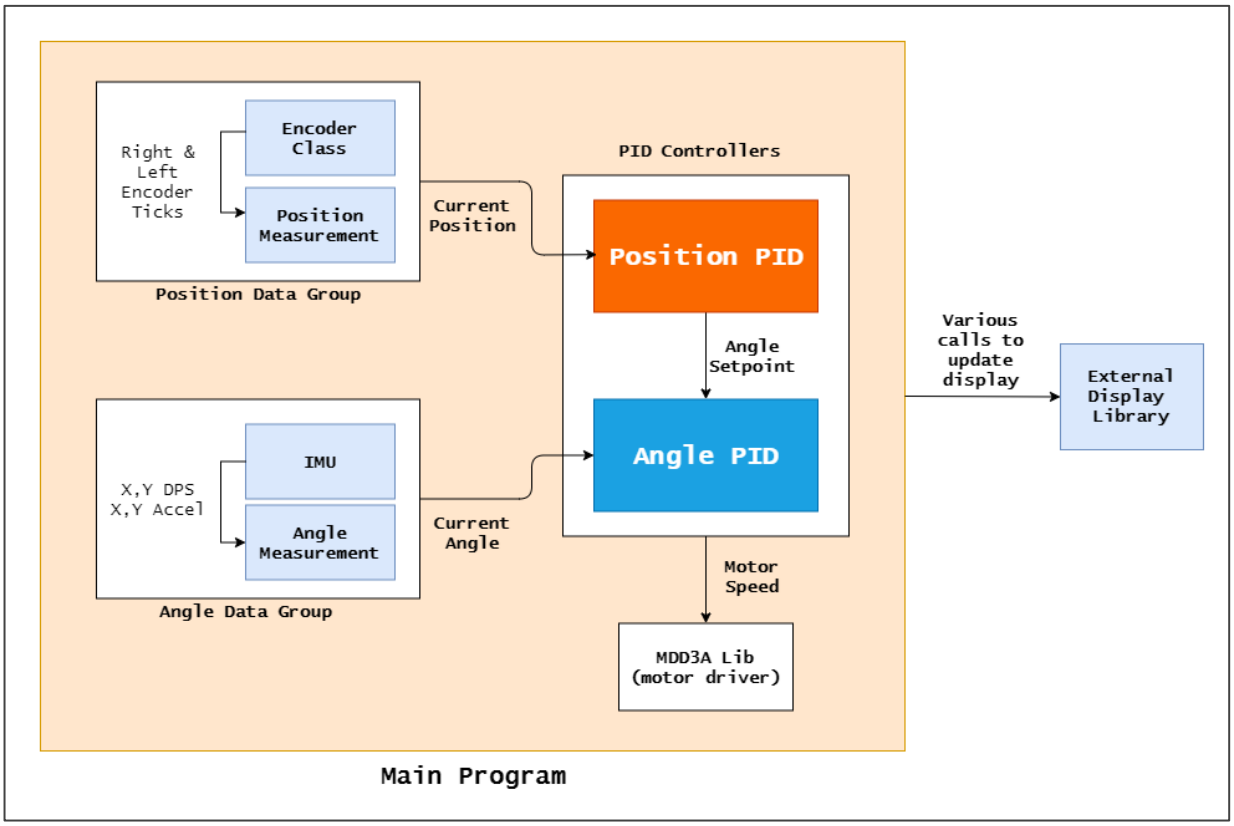
Софтуерна архитектура

Софтуерни модули

- **IMU –** Класът IMU управлява комуникацията с IMU модула. Предоставя функции, които връщат ъгловата скорост в градуса/секунда и ускорението за всички оси.
- **Encoder –** Класът Encoder съдържа кода необходим за прочитане на стойности от енкодера. Предоставя достъп до броя отброявания на енкодера както и посоката на въртене.
- **Position Measurement –** Класът Position се грижи да изчислението на позицията на база на измерванията от екнодерите. Предоставя методи, които връщат текущата позиция за всяко колело и Update метод, който се извиква от главната програма.
- **Angle Measurement –** Класът AngleMeasurement управлява изчислението и запаметяването на текущия ъгъл. Предоставя функции, които връщат текущия ъгъл на IMU модула за X и Y осите. Този клас трябва да бъде актуализиран в цикъла на главната програма
- **I2C Utilities –** Класът i2c_utils предоставя обвивка за функциите от библиотеката Wire.h. Тази обвивка е проектирана за по-лесно четене на и писане към регистри
- **Motor driver –** Класът MDD3A е интерфейса за драйвера на мотора. Предоставя единствено функцията UpdateSpeed(), която приема като аргументи скорост и посока.

Окончателна структура

Фиг. 23 показва крайната структура на софтуера. По-голяма версия на схемата може да бъде намерена на Github (Nikola, 2021) страницата на проекта.



Фиг. 23 - Software Structure

Контролер

Настройка на PID

Както беше споменато в раздела „Автоматична настройка на ПИД в MATLAB“, програмата MATLAB има автоматична функция за настройване на ПИД контролери, въпреки това, тя не може дас е използва за ПИД имплементиран на микроконтролер. Поради тази причина, се използва емпирично настройване на ПИД контролера имплементиран на микроконтролера. Когато приемливо поведение на системата е постигнато, ПИД контролера се счита за настроен.

Внедрена функционалност

Режим на балансиране

Режима на балансиране е успешно имплементиран. Системата леко трепери, но това не се отразява негативно върху функционалността. Допълнително настройване на ПИД контролера може да помогне с премахването на този проблем, друго подобрение е по-добра инсталация на IMU модула. Тези подобрения се обсъждат в раздела „Бъдещо развитие“.

Режим на задържане на позицията

Режимът за задържане на позицията също е имплементиран, но поради лоши сензорни данни не е точен колкото режима за балансиране. В раздела „Бъдещо развитие“ може да се намери повече информация за проблема и какви решения съществуват.

Тестване

Тестването е ключова стъпка от процеса на разработка и този раздел ще разгледа тестовите стратегии, оборудване и резултати от тестовете

Стратегия за тестване

Тъй като повечето компоненти от Teensy BalanceBot Mk1 са свързани с съответен програмен код, всички тестове имат смес от хардуерни и софтуерни тестове.

Тестовата стратегия е да се разработи една функционалност, например кода за работа с IMU модула и след това дас е тества дали хардуера и софтуера работят правилно.

Тестването на малки части от системата осигурява, че когато всички компоненти се съберат заедно, те ще работят както трябва. Благодарение на тази стратегия минимален брой проблеми възникнаха при свързването на компоненеите.

Оборудване за тестване

Тестовото оборудване се състои от осцилоскоп, мултимер, настолно захранване и компютър.

Осцилоскопът се използва за проверка на сигналите които се предават между компонентите, както и да се провери дали дадени изчисления се изпълняват за правилното време.

Мултимерът се използва за проверка на електрическите вериги и запоените компоненти.

Компютъра се използва за качване на код върху микроконтролера, разработка на кода, и проверка на показанията от сензорите.

Описание на тестовете

Този раздел разглежда по-подробно проведените тестове.

Тест на електрическата верига

Тестовете на електрическата верига обхващат тестване на платките и окабеляването. Единствената платка, която трябва да се провери е тази, към която се закача микроконтролера.

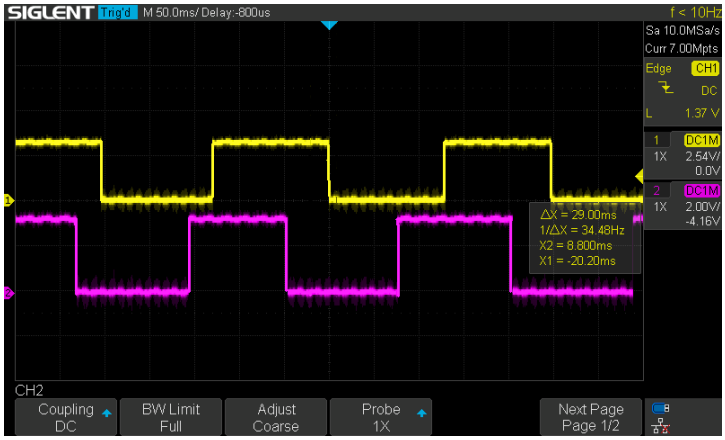
Проверката се прави с мултимера на режим „последователност“. Пробна жица се поставя в рейка, а друга се закача до крайната точка на дадения пин (например от пин за енкодера до платката на енкодера). По този начин се тества платката за микроконтролера, както и цялото окабеляване до сензора/актуатора.

Тест на енкодерите

Хардуерен тест

Теста за енкодера започва като се закачи енкодера за осцилоскопа и се захранва с 3.3V. Помощната платка се слага в поставката за енкодера и магнитите се поставят на моторите, а моторите се пускат да работят.

Ако енкодера работи правилно, на осцилоскопа трябва да се покаже квадратната вълна показана на фиг. 24.



Фиг. 24 - Encoder waveform

Софтуерен тест

Софтуерния тест се състои от закачане на енкодера за микроконтролера и преброяване завъртанията на енкодера с прост тестов код. Когато се установи, че кодът работи правилно, той се преработва и се превръща в модула за работа с енкодерите.

Тестване на екрана

Тестът за екрана се състои от закачането на пиновете на модула за микроконтролера и използвайки външната библиотека от производителя да се тества дали модула работи.

Тестване на инерционния модул IMU

Теста на инерционния модул се състои от закачането на правилните пинове на сензора към тези на микроконтролера. След това тестов код, който прочита ъглово ускорение и линейно ускорение. Ако този код работи и извежда достоверни резултати, се зарежда друг код, който използва данните от акселерометъра и жироскопа за да измери ъгъла на наклон на сензора. Сензора се накланя на ъгъл чийто размер се знае и се сравняват резултатите. Накланянето се прави с помощта на чертожен триъгълник.

Тестване на моторите и драйвера.

Моторите и драйвера за мотори се тества използвайки хардуерните бутони на драйвера. Когато се провери, че драйвера и моторите работят използвайки само бутоните, тестов код се пише за да се провери дали скоростта може да се управлява. Тестовата програма сменя скоростта от минимална на средна на висока и обратно до нула.

С осцилоскоп се проверява дали микроконтролера произвежда правилния PWM сигнал.

Бъдещо развитие

This section discusses some options for future development, what issues the current project have and how they can be solved.

Текущи проблеми

Неточни енкодери

Проблем

В момента, начина по който енкодерите са инсталирани, при едно пълно завъртане на колело, енкодера има 3 отброявания. Това означава, че точността е много лоша и има значително отместване на робота преди да има корекция.

Решение

Има няколко решения на проблема, първото е да се инсталира по-добър енкодер (с повече стъпки на завъртане) за колелата. Друго решение е да се използват правилните мотори за енкодерите, такива с удължена ос отзад. На фиг. 25 се вижда правилното монтиране на енкодера и магнитите.

Фиг. 25 - Correct encoder mounting

Монтиране на инерционния модул

Проблем

В момента IMU модула е закрепен за breadboard. Този метод създава няколко проблема. Първия е, че всички вибрации се долавят от сензора и създават шум в измерването, което се превръща в лошо поведение на системата. Друг проблем е, че модула лесно може да се наклони и да засича неправилен ъгъл, и това може да доведе до грешни команди към моторите. Третия е, че сензора е монтиран близо до драйвера за моторите, и заради честите промени на скоростта на моторите, това може да създаде електромагнитен шум, който да повлияе отрицателно на измерванията.

Решение

Решението на първия и втория проблем е да се направи специална платка, върху която да се монтира модула. След това тази платка може да се монтира на платформа направена от TPU която да абсорбира част от вибрациите. Третия проблем може да се реши като се премести модула на по-подходящо място.

Структура на роботи и сглобяване

Проблем

Сегашната структура на робота е прекалено малка за да се правят подобрения или да се добавят нови сензори.

Колелата са прекалено тънки и не предоставят достатъчно сцепление

Страничните панели се извиват при по-големи натоварвания.

Трудно се премахват батериите от отделението.

Няма добра организация на окабеляването, ако ще се добавят повече сензори, това трябва да промени.

Решение

За съжаление, за да се решат тези проблеми, роботът трябва да се проектира наново. Освен да реши сегашните проблеми, нова версия на робота ще позволи за правилно монтиране на нови сензори и актуатори.

Мотори

Проблем

Моторите, които се използват не са с добро качеството и предавките имат много люфт. Точките за монтиране на колелата се са прави и се наблюдава клатене на колелата. Няма и начин за инсталиране на енкодери върху осите на мотора.

Решение

Метални предавки, като тези предлагани от Pololu са по-добър вариант. Компанията Actobotics също произвежда добри мотори с вградени енкодери. Друга възможност е да се използват стъпкови мотори, но в този случай трябва да се добави втори драйвер или да се смени с друг модел, защото драйвера MDD3A може да контролира само един стъпков мотор.

Бъдещи подобрения

Този раздел ще разгледа някои възможни бъдещи подобрения, които мога да се направя за следващата итерация на робота.

Хардуер

LiDAR Сензор

LiDAR сензор може да бъде добавен за точно измерване на разстояния. Този модул може да бъде инсталиран на въртяща се платформа за да се постигне 360 градусово зрително поле което да се използва в SLAM алгоритъм.

Сонар

Може да се добави сонар, който да се използва за избягване на колизии, защото има по-голямо зрително поле. Въпреки че има по-лоша точност от LiDAR модула може да се комбинират данните за да се направи карта на заобикалящата среда.

Радиоуправление

Безжични модули за Bluetooth или WiFi могат да бъдат добавени за радиоуправление. LoRa модули могат също могат да се добавят и да се използват за телеметрия.

Функционалност

Създаване на графичен интерфейс

Графичен интерфейс може да се създаде за телефон или компютър. С него, роботът може да се настройва и контролира и може да се добави опция за жива видео връзка.

Добавяне на контрол за позицията

Към съществуващия код може да се добави такъв за контрол на посоката (напред, назад, ляв/десен завой). Командите за посоката могат да се изпращат от потребителския интерфейс или дистанционно.

Избягване на препятствия

Може да се добави избягване на препятствия. Този режим може да се комбинира с контрол на позицията за да помага на управляващия да избягва сблъсъци с околната среда.

Картографиране на околната среда с помощта на ROS и SLAM

ROS (Robot Operating System) може да се инсталира върху Raspberry Pi, което да се монтира на робота. С помощта на ROS, SLAM (Simultaneous Localization and Mapping) алгоритъм може да се имплементира. С такава функционалност роботът може да стане напълно автономен.

Източници

Nikola, T. (2021, 01 28). *Project Github Page*. Retrieved from Github: https://github.com/NikolaTotev/Teensy-Balance-Bot-Mk_1

Pololu. (2021, 1 18). *Pololu*. Retrieved from Pololu.com: https://www.pololu.com/product/3575

R.G. Lerner, G. T. (1991). *Encyclopaedia of Physics (second Edition)*. VHC Publishers.

Robotshop. (2021, 1 18). *Robotshop*. Retrieved from Robotshop: https://www.robotshop.com/en/microduino-self-balancing-robot-kit.html

Steven L. Brunton, J. N. (2017). *Data Driven Science & Engineering (Machine Learning, Dynamical Systems, and Control)*. Washington.