

# Robottiohjelmoinnin harjoitustyö

Tasapainottelija

Henri Vaara

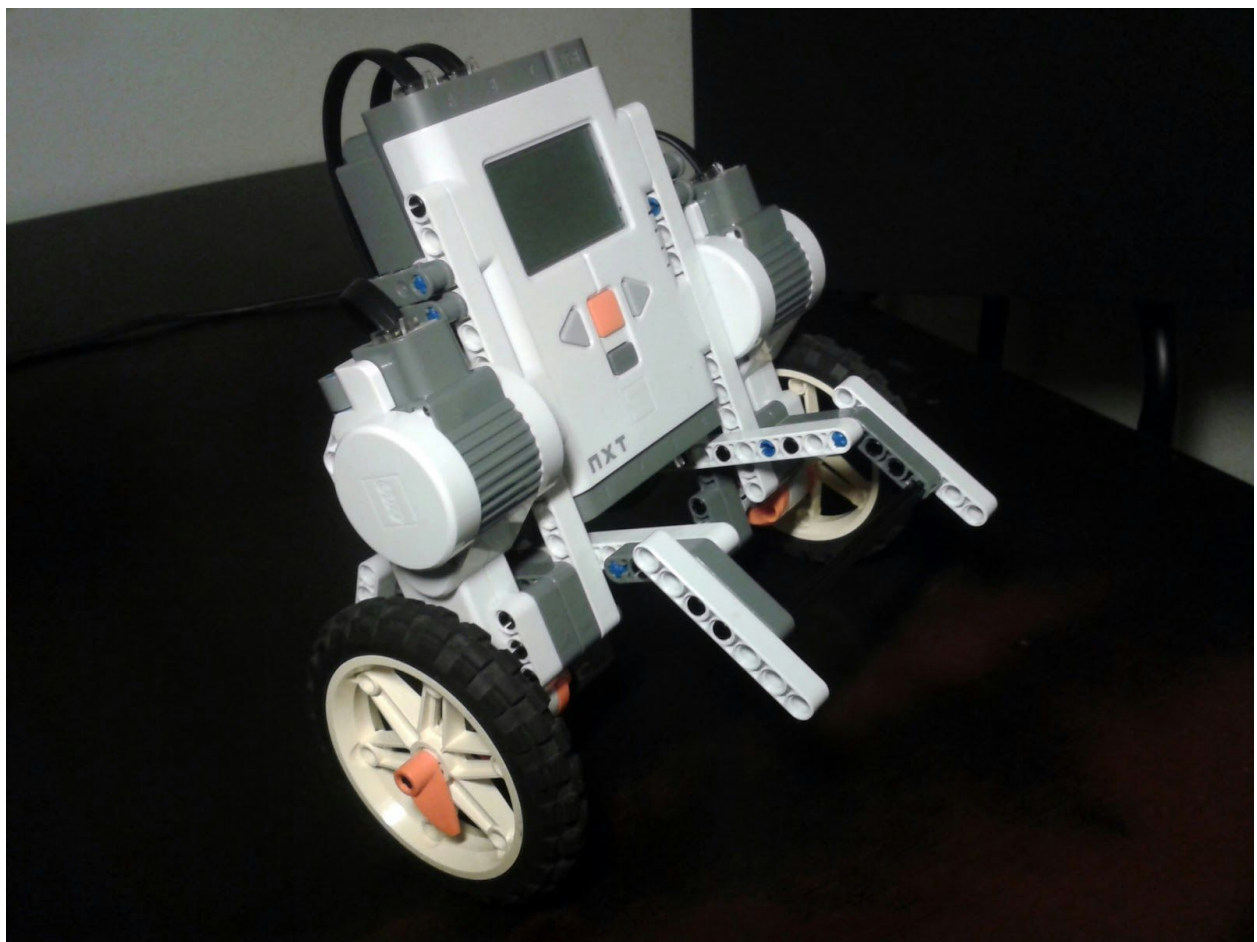
014312946

[henri.vaara@helsinki.fi](mailto:henri.vaara@helsinki.fi)

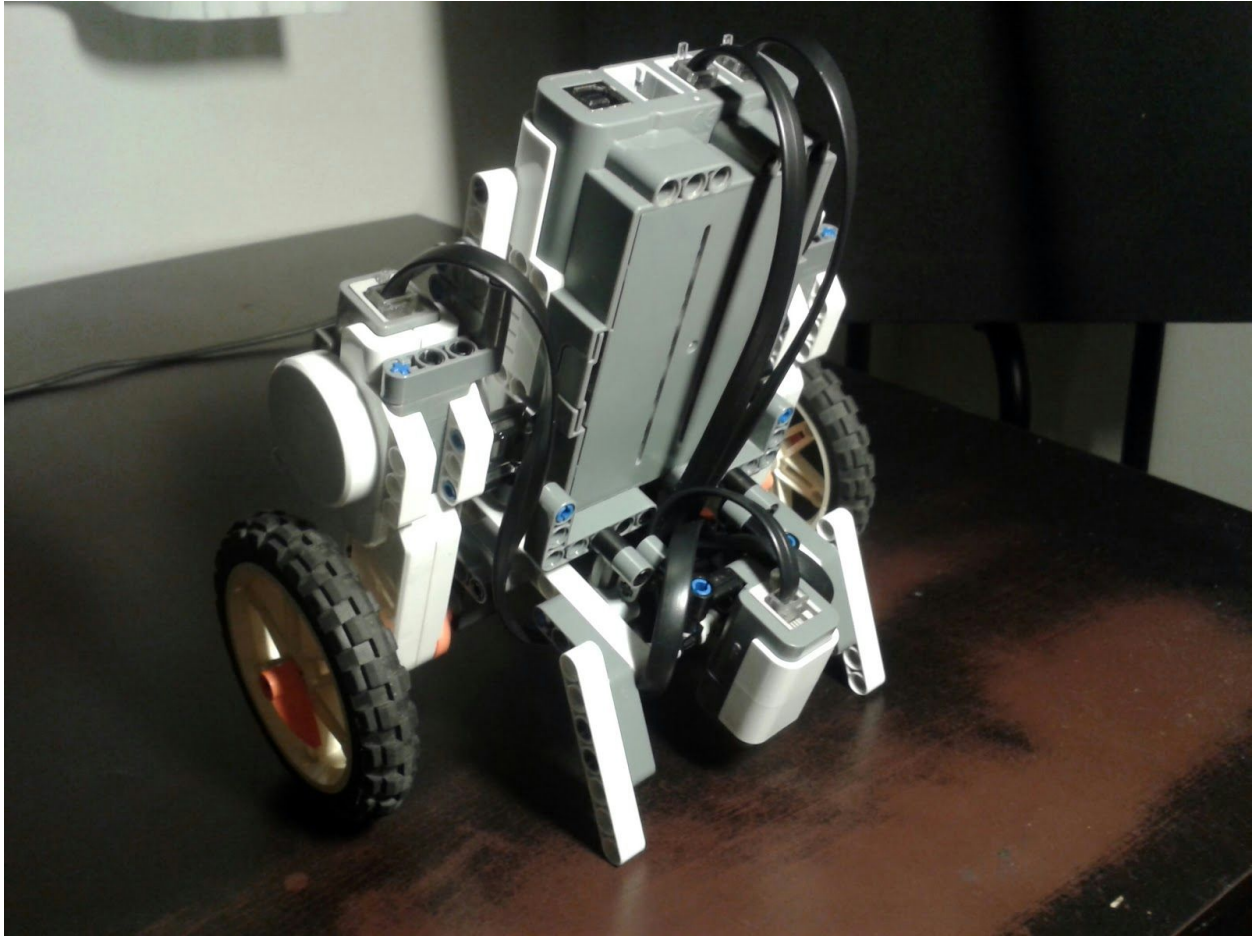
## Robotin kuvaus

Tasapainottelija on robotti joka pysyy pystyssä kahden renkaan varassa, jos segway on tuttu kapistus niin siitä saa vähän ideaa. Tasapainottelija lukee anturillaan robotin asennon, nopeuden ja kulmanopeuden ja näiden arvojen perusteella suorittaa korjausliikkeet pysyäkseen pystyssä.

## Robotin rakenne



Tasapainottelija on kaksipyöräinen robotti jolla on säädettävät "tip over" tuet suojaamassa kaatumiselta josta ei enää voi palata robotin omin voimin pystyyn. Robotti saavutti lopullisen muotonsa noin seitsemännen kehitysversion jälkeen, tämä versio on tasapainossa robotin ollessa täsmälleen pystysuorassa. Rakenne on pyritty pitämään mahdollisimman kestäväenä ja yksinkertaisena, kasattuna se ei taivu mihinkään suuntaan ja on yleisesti melko jämäkkä. Pienenä lisä hienoutena voi pitää pystysuunnassa säädettävää painopistettä.



Sensori jolla kulma mitataan on Legon color detection sensor, siitä käytetään ominaisuutta jossa led antaa kiinteän samanvahvuisen valon joka heijastuu maasta takaisin anturissa olevalle fotoresistorille, tällä saamme selvitettyä melko tarkasti mikä on anturin etäisyys maanpinnasta sekä johdettua lopulliset arvot robotin kulmasta ja kulmanopeudesta.



Renkaitten akselit on tuettu vakaajilla, jotta moottorien laakerointiin kohdistuva vääntävä voima saadaan minimoitua.

## Koodin rakenne

Koodi koostuu seitsemästä luokasta:

- Pääluokka HelloWorld hoitaa tasapaino-ohjelman käynnistymisen sekä anturin kalibroinnin robotin käynnistyessä.

- BalanceController vastaa robotin tasapaino-ohjelman toiminnallisuudesta, se lukee arvoja anturi luokalta ja käskyttää moottorien ohjaus luokkaa. Iteration metodi tekee aina yhden korjausliikkeen. Tip over cut off, eli moottoreiden sammutus kun robotti nojaa tukia vasten on annettu tämän luokan hoidettavaksi.

- AngleAcceleration luokka on tuppeli jolla siirretään kulma ja kulmanopeus SensorController2 luokasta BalanceControllerille.

-CorrectionMap luokka luo kartan josta haetaan aina kyseistä kulma ja kulmanopeus paria vastaava korjausarvo. Kun konstruktori ajetaan, lasketaan  $\text{next}$ :n muistiin hajautustaulut joilla jokaisella kulmalla -100 to 100 löytyy indeksi kartasta sekä jokaiselle kulmanopeudelle hajautustaulu jolla löytyy indeksi korjaus kartasta. Varsinainen kartta on diagonaalisesti projisoitu 21x21 taulukko jonka arvot tuotetaan  $ax^3+bx$  funktiolla. Vakioilla  $a$  ja  $b$  säädetään korjausliikkeen voimakkuutta sekä aggressiivisuutta.

-MotorController2 luokka käskyttää moottoreita ja sitä käytetään BalanceControllerista, maanopeuden korjaus toteutetaan MotorController2 luokassa, koska sillä on suoraviivaisin pääsy moottorien pyörimisnopeuteen josta maanopeus voidaan selvittää.

-MovingPointAverageFilter on liikkuvan pisteen keskiarvo suodatin jolla anturista saatuja arvoja hieman tasataan ja pyritään välttämään mahdollisten virhe lukemien aiheuttamaa kohinaa ja näin tasaamaan robotin käyttäytymistä.

-SensorController2 vastaa anturidatan käsittelystä, tällä luokalla tuotetaan kulma-arvo sekä kulmanopeus.

## Testaus

Robottia on testattu käytännössä tarkkailemalla sen käytöstä, kun se nojaa tukiinsa, moottorit pysähtyvät eikä robotti karkaa. Anturi dataa on testattu koodin debug osioilla joilla etäisyys arvosta johdetut kulmanopeus ja kulma tiedot printataan näytölle, näin on saatu todistettua, että anturin data on oikeellista.

## Rajoitukset ja tulevaisuus

Vielä en ole löytänyt lopullisia korjausarvoja  $a$  ja  $b$  joilla robotti pysyisi täysin itsenäisesti pystyssä, kolmen viikon aika ei vain riittänyt tähän. Pienellä tuella se kuitenkin toimii jo melko kiitettävästi. Mikäli aikaa olisi ollut enemmän olisi koodiin voinut tehdä lisäyksiä joilla robotti olisi osannut ponnauttaa itsensä ylös kaatumisen jälkeen. Mikäli tasapaino olisi saavutettu ja robotin olisi haluttu liikkuvan eteen ja taaksepäin sekä kääntyvän, olisi ainakin eteen- ja taakse liikkeet olleet helppo toteuttaa muuttamalla hetkellisesti muistiin kirjatun tasapainopisteen kohtaa. Muutokset valaistuksessa saavat robotin kaatumaan, joten sen on aina oltava tasaisessa valaistuksessa ja mielellään hämärässä jotta robotista muodostuneet varjot eivät häiritse anturin toimintaa.

## Käyttöohje

Suorita ohjelma, robotti aloittaa antamalla nousevan äänimerkin, tämä indikoi, että se tulee asettaa nenä ylöspäin asentoon. Seuraavaksi robotti antaa laskevan äänimerkin jonka jälkeen se tulee asettaa nenä alaspäin asentoon. Näin ääriasennot on kalibroitu ja viimeisenä yhden piippauksen jälkeen tulee etsiä tasapainopiste jonka tallentamisen jälkeen robotti aloittaa tasapaino-ohjelman suorittamisen, loppu onkin sitten ihmettelyä ja magiaa!