

Ball Balancing Robot

Nume: Radu Eduard-Andrei Grupa: 333CC

Introducere

Un robot care foloseste 4 stepper-motors pentru a inclina un plan pe care se afla un touchpad rezistiv care poate detecta pozitia unul ball bearing. Va folosi 2 algoritmi de control de tip PID pentru a muta ball bearing-ul catre centrul sau (sau orice punct de pe suprafata lui).

M-am inspirat din videoclipuri vazute online, si l-am ales deoarece am vrut ca proiectul meu sa includa un algoritm de control. Nu are o utilitate practica, dar poate fi vazut ca o jucarie anti-stres.

Descriere generală

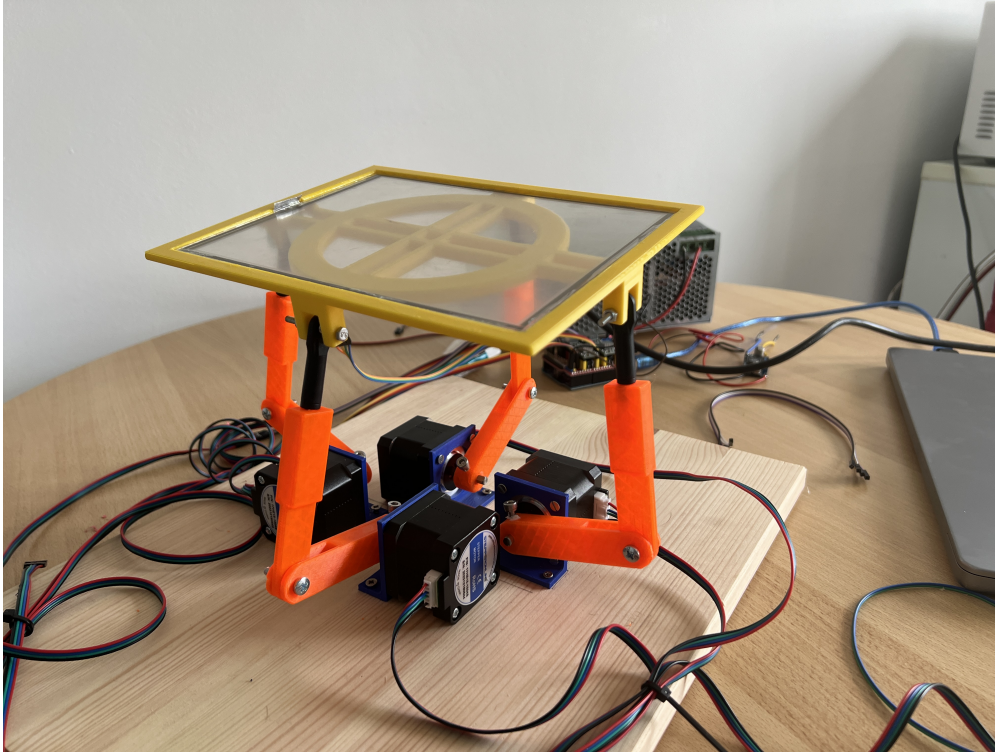
Robotul este format din piese printate 3D si consta intr-o platforma care isi poate schimba unghiul cu suprafata orizontala prin intermediul celor 4 motoare care controleaza cea 4 picioare.

Se incearca balansarea unei bile de metal pe platforma. Pentru determinarea pozitiei bilei se foloseste un touchpad rezistiv.

Se foloseste un algoritm de control de tip PID pentru determinarea unghiului optim pentru ca bila sa ajunga in centrul platformei si sa ramana acolo.



Hardware Design



Lista piese:

- Arduino uno
- 4x STEPERONLINE Stepper Motor Nema 17 Bipolar
- 4x AzDelivery A4988 Stepper Motor Driver
- 1x VSDISPLAY 8.4" Resistive Touch Sensor
- 1x Sursa DC 24V 0-20A
- 1x LM2596 DC-DC Buck Converter Step Down Modul
- 1x CNC Arduino Shield



Software Design

Pentru implementare am folosit un arduino uno R3 cu procesor atmega328p. Am folosit biblioteca AccelStepper pentru controlul mai facil al motoarelor.

Structuri folosite:

- pidX - pentru controlul unghiului dintre plan si axa OX
- pidY - pentru controlul unghiului dintre plan si axa OY
- touchpad - pentru furnizarea pozitiei bilei
- stepperController - pentru setarea celor 4 steppere in pozitii corecte rezultate din PID-uri

Algoritmi folositi:

PID

```
class PID {
public:
    PID(float kP, float kI, float kD, float target);
    void set_target(int target);
    void update(int current);
    float get_value();
private:
    float kP;
    float kI;
    float kD;

    float target;

    float proportional;
    float integral;
    float derivative;
};

PID::PID(float kP, float kI, float kD, float target) : kP(kP), kI(kI), kD(kD),
target(target) {
    proportional = 0;
    integral = 0;
    derivative = 0;
}

void PID::set_target(int target) {
    this->target = target;

    this->proportional = 0;
    this->integral = 0;
    this->derivative = 0;
}

void PID::update(int current) {
    current -= 512;
    float normalized_current = current / 512.0;
    derivative = target - normalized_current - proportional;
    proportional = target - normalized_current;
    integral += proportional;
}
```

```
float PID::get_value() {  
    return kP * proportional + kI * integral + kD * derivative;  
}
```

Rezultate Obținute

In final robotul functioneaza fiind capabil sa balanseze o bila de metal fara ca aceasta sa pice de pe platforma pe o perioada lunga de timp. Totusi merita subliniate impreciziile mecanice pe care robotul le are. Cea mai importanta este ca piesele printate 3d nu au un sistem de prindere cu motoarele adecvat, ceea ce duce la deviatii de unghi ale bratelor in timpul utilizarii. Consecinta acestui luctu este incapacitatea de a face punctul de echilibru al robotului constant. O a doua problema intampinata este un defect de fabricatie al senzorului touchscreen care face ca uneori citirea sa fie complet eronata.

Download

Am adaugat codul pentru algoritmul de control si modelele 3mf ale pieselor necesare in acest repository de github:<https://github.com/radueduard/BallBalancingRobot>

Bibliografie/Resurse

- <https://www.arduino.cc/reference/en/libraries/accelstepper/>
- <https://www.arduino.cc/reference/en/libraries/adafruit-touchscreen/>

[Export to PDF](#)

From:
<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2024/tdicu/eduard_andrei.radu



Last update: **2024/05/27 13:55**