

# Self Balancing Robot

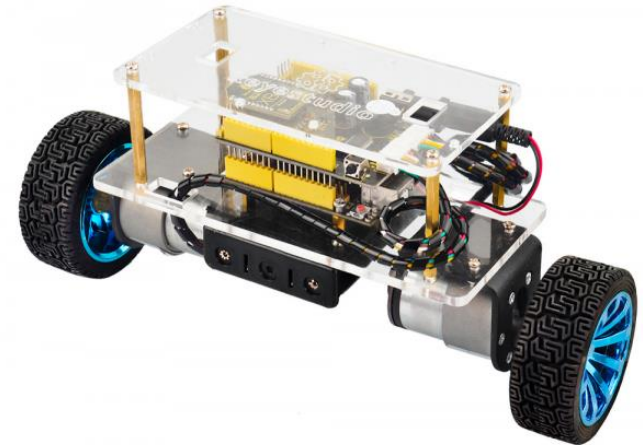
---

ALVANDI DAMANSYAH || 1103192191

# Introduction

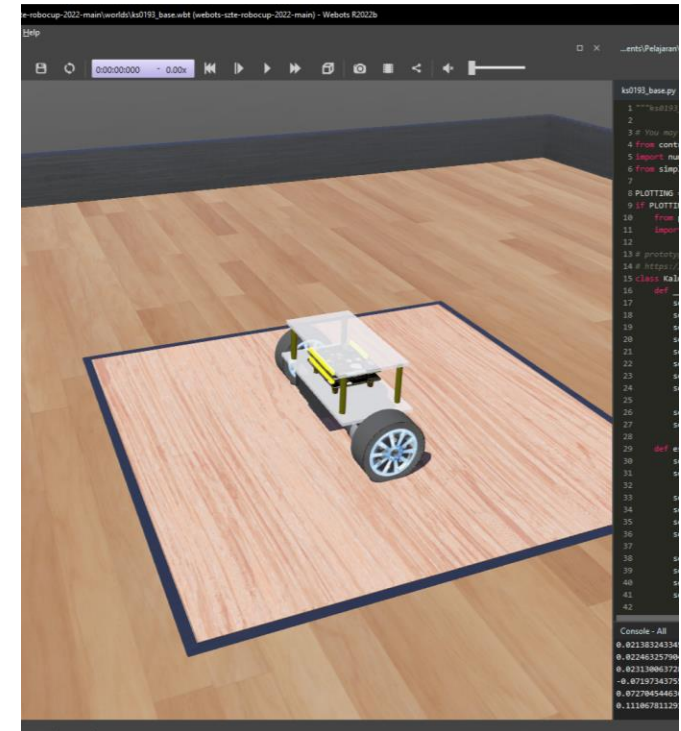
---

robot ks0193 merupakan model robot yang digunakan untuk melakukan percobaan self-balancing. Pada model ini robot hanya dilengkapi dengan dua roda dan sensor yang dimana robot akan menyesuaikan dirinya dengan giroskopnya dan kecepatan yang dibutuhkannya untuk menyeimbangkan dirinya.



# Introduction

Webot merupakan sebuah aplikasi open-source yang digunakan untuk mensimulasikan robot. Webot menyediakan lingkungan untuk model, program, serta mensimulasikan robot. Bahasa program yang dapat digunakan C/C++, Java, Python, dan MATLAB™.





# Overview

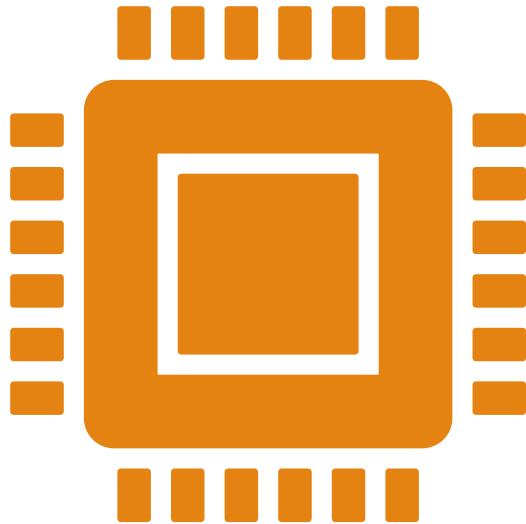
---

Pada webot akan menggunakan webots-szte-robocup-2022, dengan robot, lingkungan, dan program yang telah disediakan. Webot akan mensimulasikan robot ks0193 dengan :

- Robot akan mencoba mencoba self-balancing ditempat
- Robot akan mencoba self-balancing sambil berjalan
- Robot akan mencoba self-balancing dengan beban yang disediakan

# Architecture

---



## **Controllers**

Controllers merupakan folder yang berisi program yang mengontrol pergerakan dari robot ks0193 yang nantinya akan disimulasikan.

## **Libraries**

Libraries merupakan folder berisi gambar yang digunakan untuk membantu membentuk fisik robot.

## **Protos**

Protos merupakan folder yang berisi ekstension file untuk membentuk fisik robot, dan trod.

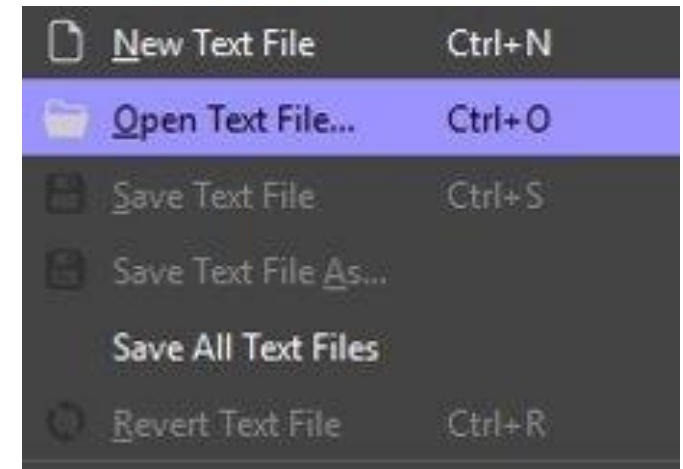
## **Word**

Word merupakan folder yang berisi lingkungan yang dapat dijalankan pada webot

# Webot

---

Selanjutnya untuk membuka kontrol program dapat dilakukan dengan membukan open text file pada toolbar file



# Kalman filtering

---

Pada kontrol robot yang digunakan menggunakan kalman filtering.

Kalman filtering merupakan algoritma yang menggunakan persamaan keadaan sistem linier untuk memperkirakan keadaan sistem melalui masukan dan keluaran sistem. Dikarenakan askelerometer sensitif terhadap getaran mekanis yang dapat mempengaruhi error.

# PID

---

Proporsi deviasi, Integral, dan Diferensial digabungkan secara linear untuk membentuk besaran kontrol, kemudian mengontrol objek dengan besaran kontrol ini. hal ini disebut pengontrol PID.

- Proporsi deviasi digunakan untuk bereaksi terhadap momen deviasi
- Integral digunakan untuk menghilangkan deviasi sistem
- Diferensial digunakan untuk mengurangi overshoot, mengatasi osilasi, dan menstabilkan sistem, terutama untuk sistem orde tinggi, yang mempercepat kecepatan pelacakan sistem



# Self-Balancing on Spot

---

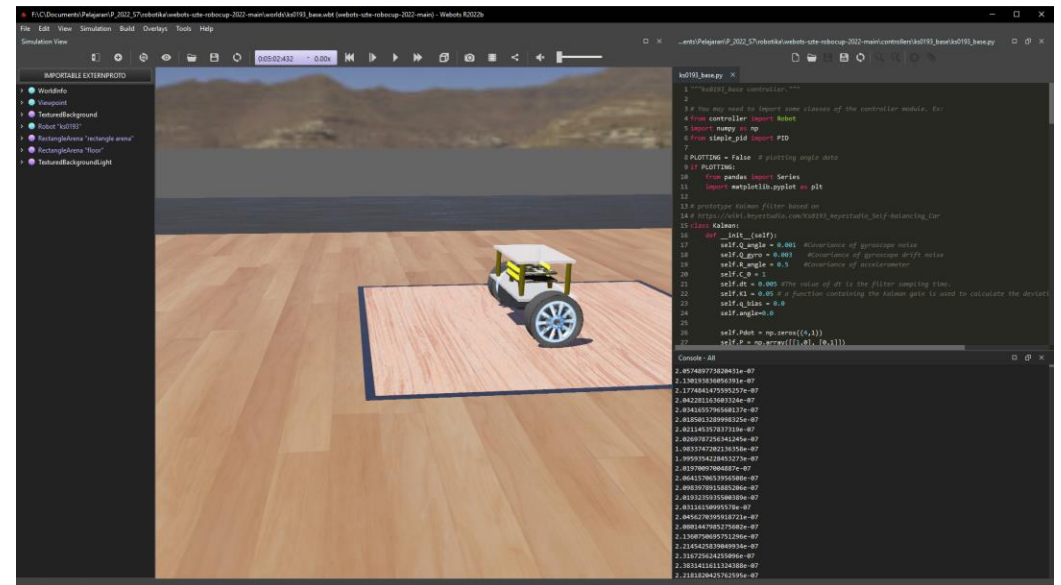
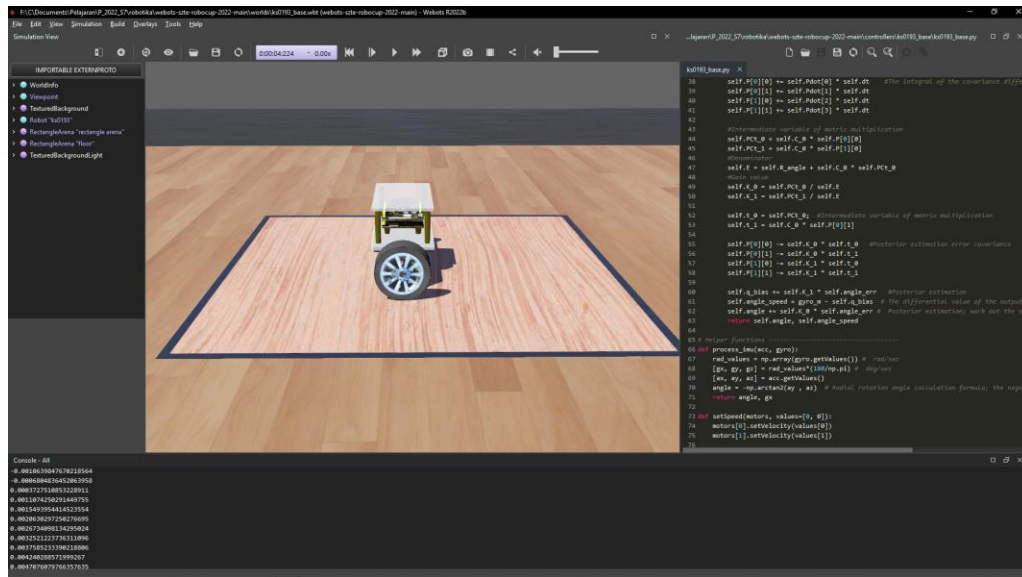


file lingkungan yang dijalankan **ks0193\_base.wbt** dan kontrol menggunakan **ks0193\_base.py**



Saat simulasi dijalankan robot memulai dengan sudut  $-0.001$  menuju ke sudut positif, pada sudut 6.x robot mendapatkan puncak kestabilan dan tetap stabil dalam 5 jam simulasi

# Self-Balancing on Spot



# Self-Balancing While manouvering

---

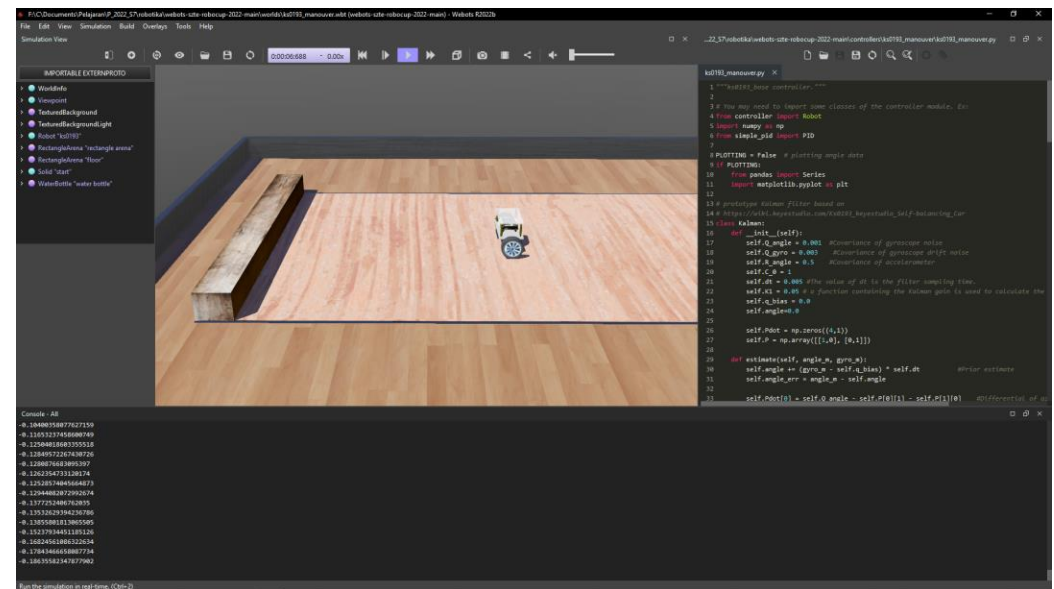
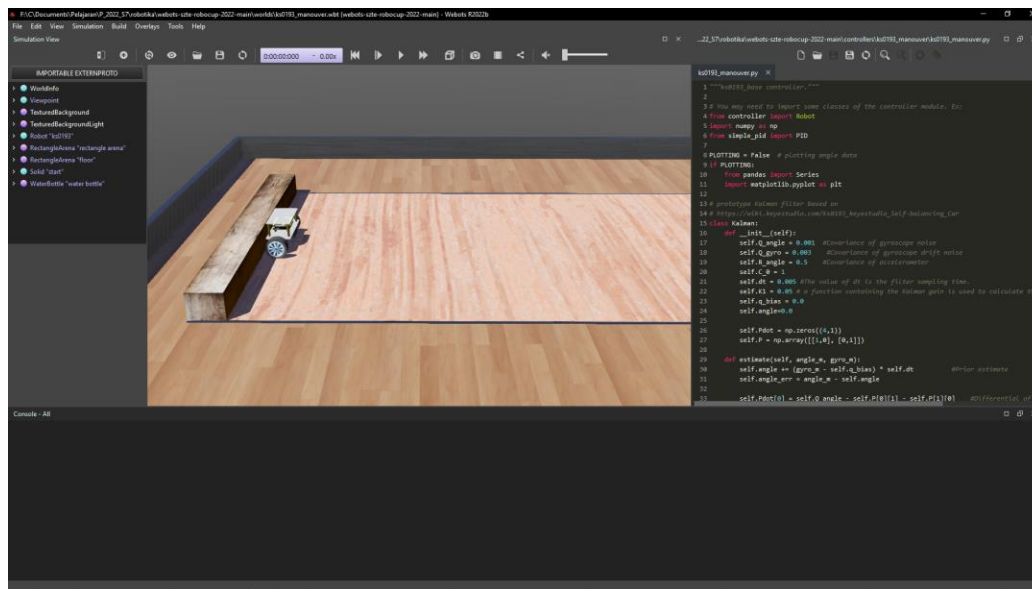


file lingkungan yang dijalankan **ks0193\_manouver.wbt** dan kontrol menggunakan **ks0193\_manouver.py**



Dalam simulasi robot berhasil berjalan dengan stabil dengan sudut kestabilan  $-0.1x$  hingga bertabrakan dengan botol

# Self-Balancing While manouvering

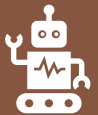


# Self-Balancing on Spot with Weight

---



file lingkungan yang dijalankan **ks0193\_weight.wbt** dan kontrol menggunakan **ks0193\_weight.py**



Dalam simulasi robot ks0193 diberikan beban trod, robot mendapatkan puncak sudut kestabilan pada  $-13$  dan tetap stabil dalam 2 jam simulasi

# Self-Balancing on Spot with Weight

