

# Program Robot Sepak Bola Manual

Figo Arzaki Maulana - 5022221041

Rabu, 17 Maret 2025

## 1 Penjelasan Program Robot Sepak Bola

Program ini merupakan implementasi sistem kendali robot diferensial Pioneer P3DX yang digunakan dalam simulasi CoppeliaSim.

### 1.1 Parameter Robot

Robot memiliki dua parameter fisik utama yaitu radius roda ( $R$ ) sebesar 0,0975 meter dan jarak antar roda ( $L$ ) sebesar 0,381 meter. Program mengimplementasikan sistem debouncing dengan waktu tunda 0,1 detik untuk mencegah input berulang yang tidak diinginkan saat tombol ditekan.

### 1.2 Sistem Kendali

Sistem kendali robot menggunakan inverse kinematics untuk mengkonversi kecepatan linear dan angular yang diinginkan menjadi kecepatan rotasi masing-masing roda. Fungsi `inverseKinematics` menggunakan matriks inverse untuk menghitung kecepatan roda kanan dan kiri berdasarkan input kecepatan yang diberikan.

### 1.3 Implementasi Kontrol

Kontrol robot dilakukan melalui keyboard dengan tombol panah. Tombol panah atas dan bawah mengatur kecepatan linear (maju/mundur) dengan perubahan 0,1 unit setiap penekanan, sedangkan tombol panah kiri dan kanan mengatur kecepatan angular (berbelok) dengan perubahan yang sama. Program membatasi kecepatan linear maksimum pada 1 m/s dan kecepatan angular maksimum pada 0,5 rad/s untuk menjaga stabilitas pergerakan robot. tombol reset (R) dapat mengembalikan kecepatan ke nilai nol.

## 2 Diagram Kontrol

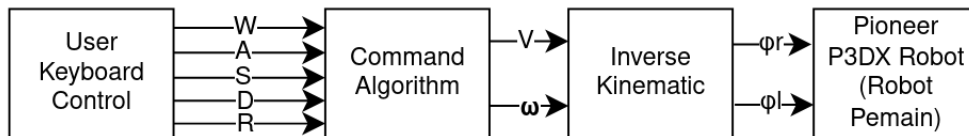


Figure 1: Diagram Kontrol Robot

### 3 Kode Program

```
1
2 from coppeliasim_zmqremoteapi_client import RemoteAPIClient
3 import keyboard
4 import time
5 import numpy as np
6
7 R = 0.0975 # radius of the wheel
8 L = 0.381 # distance between the wheels
9 DEBOUNCE_TIME = 0.1 # 100ms debounce time
10 last_key_press = {
11     "up": 0,
12     "down": 0,
13     "left": 0,
14     "right": 0,
15     "r": 0
16 }
17
18 def inverseKinematics(v, w):
19     inverse_matrix = np.linalg.inv(np.array([[R/2, R/2], [R/(2*L), -R/(2*L)]]))
20     return np.dot(inverse_matrix, [v, w])
21
22 def getMotorsHandle():
23     motorRightHandle = sim.getObject("/Robot_Pemain/rightMotor")
24     motorLeftHandle = sim.getObject("/Robot_Pemain/leftMotor")
25     return (motorRightHandle, motorLeftHandle)
26
27 def setRobotMotion(motorsHandle, veloCmd):
28     _ = sim.setJointTargetVelocity(motorsHandle[0], veloCmd[0])
29     _ = sim.setJointTargetVelocity(motorsHandle[1], veloCmd[1])
30     return
31
32 client = RemoteAPIClient()
33 sim = client.require("sim")
34
35 sim.setStepping(False)
36 sim.startSimulation()
37
38 motors_handle = getMotorsHandle()
39
40 cmd_vel = [0.0, 0.0]
41 prev_cmd_vel = [0.0, 0.0]
42 wheel_vel = [0.0, 0.0]
43 lin_vel_lim = 1
44 ang_vel_lim = 0.5
45
46 while True:
47     if keyboard.is_pressed("esc"):
48         break
49
50     current_time = time.time()
51     if keyboard.is_pressed("up") and (current_time - last_key_press["up"]) > DEBOUNCE_TIME:
52         cmd_vel[0] += 0.1
53         last_key_press["u"] = current_time
54     if keyboard.is_pressed("down") and (current_time - last_key_press["down"]) > DEBOUNCE_TIME:
55         cmd_vel[0] -= 0.1
56         last_key_press["s"] = current_time
57     if keyboard.is_pressed("left") and (current_time - last_key_press["left"]) > DEBOUNCE_TIME:
58         cmd_vel[1] += 0.1
```

```

59     last_key_press["a"] = current_time
60     if keyboard.is_pressed("right") and (current_time - last_key_press["right"]) >
DEBOUNCE_TIME:
61         cmd_vel[1] -= 0.1
62         last_key_press["d"] = current_time
63     if keyboard.is_pressed("r") and (current_time - last_key_press["r"]) > DEBOUNCE_TIME:
64         cmd_vel = [0.0, 0.0]
65         last_key_press["r"] = current_time
66
67     if cmd_vel[0] > lin_vel_lim:
68         cmd_vel[0] = lin_vel_lim
69     if cmd_vel[0] < -lin_vel_lim:
70         cmd_vel[0] = -lin_vel_lim
71     if cmd_vel[1] > ang_vel_lim:
72         cmd_vel[1] = ang_vel_lim
73     if cmd_vel[1] < -ang_vel_lim:
74         cmd_vel[1] = -ang_vel_lim
75
76     wheel_vel = inverseKinematics(cmd_vel[0], cmd_vel[1])
77
78     if cmd_vel != prev_cmd_vel:
79         print(f"cmd_vel: {cmd_vel}, wheel_vel: {wheel_vel}")
80
81     setRobotMotion(motors_handle, wheel_vel)
82     prev_cmd_vel = cmd_vel.copy()
83
84
85     sim.stopSimulation()
86     print("\nProgram Ended\n")

```