

IMPLEMENTASI SISTEM NAVIGASI DENGAN METODE *PROPORTIONAL INTEGRAL DERIVATIVE* (PID) PADA ROBOT WALL FOLLOWER

Akhmad Irsyad^{*1}, Isnawaty², Rizal Adi Saputra³

^{*1,2,3}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari

e-mail : ^{*1}sentinel_irsyad@hotmail.com, ²isna.1711@gmail.com, ³rizaladisaputraa@gmail.com

Abstrak

Pada umumnya sistem kendali robot *wall follower* menggunakan kontroler PID (*Proportional Integral Derivative*). Hasil output dari penggunaan sistem kontrol jenis ini terkadang dapat membuat pergerakan robot menjadi tidak stabil dan kaku jika diberikan nilai acuan (konstanta) yang tidak sesuai. Itu disebabkan karena apabila diberi nilai konstanta yang tidak sesuai, maka respon dari robot jika ada gangguan (*Disturbance*) dapat menyebabkan nilai *overshoot* yang tinggi. Agar dapat mengatasi permasalahan ini, di perlukan suatu persamaan Osilasi untuk mengatur seluruh nilai konstanta PID, Persamaan Osilasi disini berfungsi untuk membantu menentukan nilai konstanta PID untuk memperkecil *overshoot*. Dari metode ini didapatkan hasil yaitu pergerakan robot semakin stabil dengan *overshoot* yang kecil.

Kata kunci—*Wall follower, Proportional Integral Derivative, Arduino*

Abstract

Generally the *wall follower* robotic control system uses controller PID (*Proportional Integral Derivative*). The output of the use of this type of control system can sometimes make the robot's movement unstable and stiff when given unmatched reference values (constants). That is because if given the value of money constant is not appropriate, then the response of the robot if there is disturbance can cause high overshoot value. In order to solve this problem, an oscillation equation is required to set all PID constants values. The Oscillation Equation here serves to help determine the value of the PID constant to minimize overshoot. From this method obtained the result of the movement of the robot more stable with a small overshoot.

Keywords—*Wall follower, Proportional Integral Derivative, Arduino*

1. PENDAHULUAN

Robot merupakan suatu alat yang mampu bergerak pada suatu lingkungan dengan tingkat kemampuan tertentu [1]. Robot memiliki kemampuan untuk melakukan tugas yang sederhana hingga tugas yang kompleks. Tugas sederhana yang dapat dilakukan robot diantaranya bergerak mengikuti garis, menyelesaikan labirin, dan menari. Sedangkan tugas kompleks yang dapat dilakukan robot diantaranya merakit komponen kendaraan, membersihkan limbah beracun, dan melakukan penyelamatan. Untuk melakukan tugas-tugas tersebut, robot memerlukan sistem

navigasi yang baik. Telah banyak ditemukan metode-metode yang berkaitan dengan sistem navigasi. Diantaranya yang sering digunakan adalah kontroler *Proportional Integral Derivative* (PID), *Fuzzy Logic*, algoritma genetika, jaringan syaraf tiruan dan sistem pakar. Semua metode diatas memiliki kelebihan dan kekurangan yang berbeda-beda sesuai dengan kegunaanya.

PID adalah gabungan dari metode *Proportional* (P), *Integral* (I) dan *Derivative* (D), Pengontrolan dilakukan dengan penentuan masukan berupa jarak yang diinginkan terhadap dinding (setpoint). Selanjutnya data berupa jarak diolah menggunakan mikrokontroler yang

menghasilkan sinyal untuk mengontrol robot, sehingga dapat menentukan keluaran posisi jarak yang terukur dari robot, kemudian keluaran tersebut melalui proses umpan balik dimana kesalahan ditunjukkan dengan selisih antara input (masukan) dan respon keluaran. Pada umumnya sistem navigasi dari robot *wall follower* adalah menggunakan kontroler *Proportional Integral Derivatif* (PID). Kekurangan dari sistem kontrol ini adalah, jika diberikan nilai acuan (nilai konstanta) yang tidak tepat maka akan menyebabkan pergerakan robot yang semakin kacau. Dari permasalahan yang demikian diperlukan metode kontrol yang lebih baik agar nantinya pergerakan robot dapat semakin halus.

Kecerdasan buatan merupakan salah satu cara untuk memberikan kemampuan navigasi pada robot. Kemampuan navigasi menggunakan kecerdasan buatan dapat dikaitkan dengan kemampuan sensor untuk menerima masukan dari lingkungan [1]. Masukan dari lingkungan bersifat tidak pasti sehingga pemodelan yang sederhana tidak cukup mewakili. Pemodelan menggunakan fuzzy logic dapat dijadikan alat untuk menangani ketidakpastian dan representasi pengetahuan.

Penelitian yang pernah dilakukan tentang implementasi metode *Proportional Integral Derivative* (PID) pada navigasi robot *wall follower* memiliki kekurangan yaitu kurang mulusnya pergerakan robot, sering menabrak dinding, dan kurang stabil [1].

Berdasarkan latar belakang yang telah diuraikan diatas, maka penulis mengambil topik penelitian dengan judul “Implementasi Sistem Navigasi Robot Wall follower Dengan Metode *Proportional Integral Derivative* (PID) “.

2. METODE PENELITIAN

2.1 Robot

Robot adalah perangkat mekanik yang dapat dikendalikan oleh perangkat lunak yang menggunakan sensor untuk memandu satu atau lebih efektor melalui gerakan terprogram dalam suatu ruang kerja dalam hal untuk manipulasi obyek fisik [2]. Salah satu jenis robot adalah mobile robot. Mobile robot adalah robot yang memiliki mekanisme penggerak berupa roda (*wheel*) dan atau kaki (*leg*), untuk dapat berpindah tempat dari suatu tempat ke tempat yang lain.

Robot mobil diklasifikasikan menjadi dua yaitu menurut lingkungan tempat robot tersebut bekerja dan alat yang digunakan untuk bergerak. Berdasarkan lingkungan tempat robot tersebut bekerja, robot mobil terbagi menjadi empat macam: robot yang bekerja di atas permukaan tanah (*land robot*), robot udara yang biasa disebut *unmanned aerial vehicle* (UAV), *autonomous underwater vehicles* (AUVs), dan Robot yang bekerja pada lingkungan kutub, robot yang berkerja pada kondisi permukaan tanah yang dilapisi es (*polar robots*). Sedangkan berdasarkan alat yang digunakan untuk bergerak, robot mobil terbagi menjadi robot berlengan atau berkaki-lengan atau kaki menyerupai manusia (*android*) ataupun hewan, robot beroda *Wheeled Mobile Robot* (WMR) [3].

Robot *wall follower* adalah suatu jenis robot beroda yang memiliki sensor untuk mendeteksi suatu dinding dengan sirkuit tertentu kemudian bergerak menelusuri dinding tersebut dengan modus tertentu, robot ini biasa digunakan dalam bidang industri tertentu yang dianggap berbahaya atau beresiko tinggi jika manusia melakukan hal tersebut atau bahkan justru manusia tidak mampu untuk melakukan hal tersebut [4].

2.2 Proportional Integral Derivative

Proportional Integral Derivative (PID) merupakan *controller* yang paling umum digunakan dalam dunia industri. Sistem PID biasanya digunakan dalam suatu sistem kontrol yang memiliki umpan balik (*feedback*). Controller ini terdiri dari 3 elemen konstanta yaitu *Proportional, Integral dan Derivative*. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu plant [5].

1. Proportional

Efek kontroler propotional pada suatu sistem adalah sebagai berikut:

- Jika nilai kecil, kontroler proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.
- Jika nilai besar, respon sistem menunjukkan semakin cepat mencapai keadaan yang stabil tetapi uga memungkikan motor beputar di *atas set point*. Persamaan (1) menunjukkan sinyal keluaran *control* propotional.

$$U = K_p \cdot e(t) \tag{1}$$

Keterangan:

U = sinyal keluaran *control*

K_p = *Gain controller*

$e(t)$ = *error*

2. *Integral*

Kontrol *Integral* dapat memperbaiki sekaligus menghilangkan *overshoot*, namun pemilihan konstanta integral yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem.

Karena karakteristik yang menyebabkan responnya menjadi lambat, pada umumnya konstanta *integral* dipasangkan dengan proporsional menjadi *controller* PI [5]. Persamaan (2) menunjukkan sinyal keluaran *control integral*.

$$U = \frac{1}{T_i} K_i \int_0^t e(t) \tag{2}$$

Keterangan :

U = sinyal keluaran *control*

K_i = *gain controller*

T_i = *integral time*

$e(t)$ = *error*

3. *Derivative*

Pengaruh kontrol derivatif pada sistem adalah:

- a. Memberikan efek redaman pada sistem yang berosilasi sehingga bisa memperbesar pemberian nilai.
- b. Memperbaiki respon transien, karena memberikan aksi saat ada perubahan *error*.
- c. *Derivative* hanya berubah saat ada perubahan *error*, sehingga saat ada error statis *Derivative* tidak beraksi. Sehingga *Derivative* tidak boleh digunakan sendiri. Persamaan (3) menunjukkan sinyal keluaran *control derivative*.

$$U = \frac{1}{T_d} K_d \int_0^t e(t) \tag{3}$$

Keterangan :

U = sinyal keluaran *control*

K_d = *gain controller*

T_d = *derivatif time*

2.3 Persamaan Osilasi

Metode ini didasarkan pada reaksi sistem untai tertutup yaitu *Plant* disusun *serial* dengan pengontrol PID. Pertama parameter parameter integrator diatur tak berhingga dan parameter derivative diatur nol ($T_i = \infty$; $T_d = 0$). Parameter proporsional kemudian dinaikkan secara bertahap, mulai dari nol sampai mencapai harga yang mengakibatkan reaksi sistem berosilasi. Reaksi sistem harus berosilasi dengan magnitud tetap (*Sustain oscillation*). Tabel 1 menunjukkan Persamaan Osilasi.

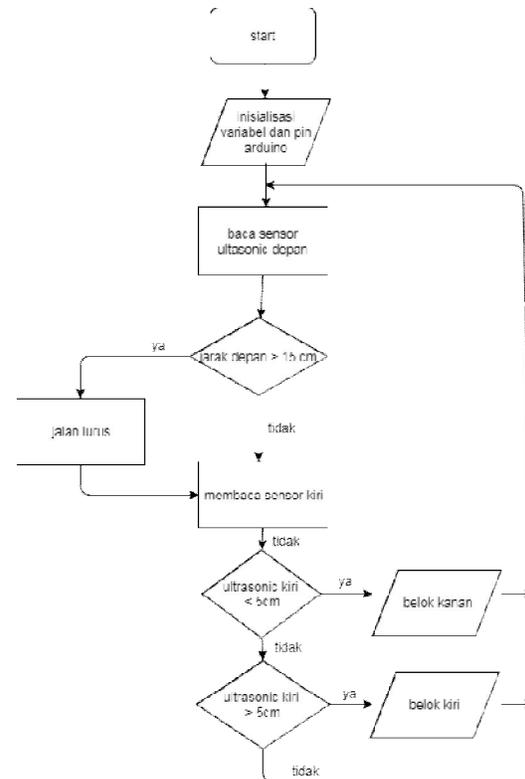
Tabel 1 Persamaan Osilasi

Tipe Pengontrol	K_p	T_i	T_d
P	0,5 K_u	-	-
PI	0,45 K_u	$\frac{1}{2} P_u$	-
PID	0,6 K_u	0.2 P_u	0,125 P_u

3. HASIL DAN PEMBAHASAN

3.1 *Flowchart*

Gambar 1 menunjukkan *Flowchart* Robot *Wall Follower*.



Gambar 1 *Flowchart* Robot *Wall Follower*

3.2 Pengujian Sistem

Pada tahap ini dilakukan pengujian terhadap kecepatan robot dengan metode PID. Dengan menempatkan robot pada arena yang telah ditentukan dan mengambil waktu yang di tempuh untuk melakukan satu putaran sebanyak 10 sample, dengan $K_u = 16$ dan $P_u = 6$. Tabel 2 menunjukkan hasil pengujian kecepatan dengan metode PID.

Tabel 2 Pengujian kecepatan dengan metode PID

No	Waktu (second)	Menabrak Dinding	Kecepatan rata-rata(cm/s)
1.	46	0	10.86957
2.	49	1	10.20408
3.	46	0	10.86957
4.	45	0	11.11111
5.	48	1	10.41667
6.	45	0	11.11111
7.	47	0	10.6383
8.	49	1	10.20408
9.	50	1	10
10.	49	1	10.20408
Rerata	47.3	0.5	10.58411

Setelah mendapatkan hasil pengujian kecepatan dengan metode PID, selanjutnya dicari rerata dari keseluruhan pengujian kecepatan yang kemudian di dapatkan bernilai 10.58411 cm/s, dengan waktu rata-rata yang di perlukan untuk menyelesaikan satu putaran 47.3, dan persentase menabrak dinding sebesar 50%.

4. KESIMPULAN

Dari Penelitian dan pembahasan Sistem Navigasi Dengan Metode PID Pada Robot Wall Follower, dapat ditarik kesimpulan sebagai berikut:

1. Penelitian ini telah berhasil mengembangkan sistem navigasi robot wallfollower yang berhasil berjalan dengan menghindari dinding. Hasil pengujian menunjukan bahwa persentase keberhasilan robot dalam menelusuri dinding ialah 50%.
2. Robot *wall follower* merupakan salah satu bentuk dari robot mobile. Salah satu ciri dari robot mobile tersebut adalah mempunyai aktuator berupa roda untuk

menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain.

3. metode PID dapat menghasilkan kecepatan robot *wall follower* 47,3 detik untuk melakukan satu putaran.

5. SARAN

Beberapa saran untuk pengembangan lebih lanjut dari robot ini yaitu :

1. Untuk pengembangan robot *wall follower* selanjutnya sebaiknya menggunakan Arduino atau rangkaian yang lebih baik agar dapat menggunakan lebih banyak sensor ultrasonik.
2. Untuk pengembangan robot *wall follower* selanjutnya bisa menggunakan metode lain seperti Jaringan Saraf Tiruan, Algoritma Genetika atau Sistem Pakar.

DAFTAR PUSTAKA

- [1] Carelli, Eduardo. Oliveira, Freire. 2003. *Robotics and Autonomous Systems*. North-Holland. Belanda.
- [2] Schilling, K. 2000. *Remote Control of mobile robot for emergency*. SPIE conference Unmanned ground vehicle technology IV. Orlando.
- [3] Walter, William Grey. 2006. *Robot Autonomous*. www.wikipedia.org/wiki/mobile_robot. Diakses 20 februari 2017
- [4] Primadi, Ardian. 2010. *Unjuk Kerja Robot Wall follower Dengan Menggunakan Sensor Ultrasonik*. Skripsi. universitas pendidikan Indonesia. Bandung
- [5] Ogata, K. 2002. *Modern Control Engineering Fourth Edition*. Aeezh. Tehran