Novateur Publications
International Journal of Innovations in Engineering Research and Technology [IJIERT]
ISSN: 2394-3696 Website: ijiert.org
Volume 11, Issue 03, March-2024

AUTOMATIC CONTROL OF A WHEELED ROBOT

Ratsimbalison Ralaitsitohaina Tafitasoa Carlos,
Doctoral Reception Team in Telecommunication,
Automatic, Signal and Image, Antananarivo, Madagascar
karylaosy@gmail.com

Ratsimba Mamy Nirina
Doctoral Reception Team in Telecommunication,
Automatic, Signal and Image, Antananarivo, Madagascar

Ramafiarisona Hajasoa Malalatiana
Doctoral Reception Team in Telecommunication,
Automatic, Signal and Image, Antananarivo, Madagascar

Article History: *Received on: 13/12/2023*
Accepted on: 21/02/2024

DOI: *https://doi.org/10.26662/ijiert.v11i3.pp1-6*

Abstract
This work aims to automatically control a wheeled robot from a specific location A to another B. We assume the robot’s trajectory is straight (the line (AB)). To solve this problem, we need the coordinates of point A and point B, i.e., the position sensor's state. A microcontroller must then process this information. Next, the variation in coordinates mean displacement, and this variation depends on speed, so we also interest in speed here. Finally, for the robot to arrive at point B, its direction of movement must be the same as the direction of (AB). The direction depends on the orientation angle of the moving robot.

Keywords: Automatic control, Robot, Trajectory, Sensors, Microcontroller, PID, etc.

Introduction

Process control is a fundamental objective in the field of automation. The most crucial aspect of control is error correction. Several techniques can be used to control wheeled robots better. The PID controller is the easiest to implement and can achieve reasonable control if the parameters are correctly adjusted. However, its performance is not sufficient, and it is inefficient for high-order and fractional systems. We're going to use the fractional-order PID controller to gain the advantages of the PID controller and also correct its shortcomings.

Fractional order PID structure

The output equation of the fractional-order PID corrector in the time domain is given by:

$$u(t)=K_p\left(e(t)+\frac{1}{T_i}D^{-\lambda}e(t)+T_dD^{\mu}e(t)\right) \qquad (1)$$

The fractional PID corrector is implemented in conventional unit feedback control systems, as shown in Figure 1

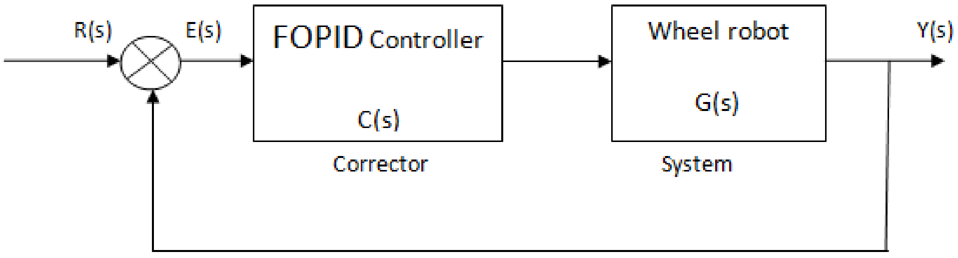

Figure 1:PIDOF control system with unitary feedback

Where u(t) is the control signal and e(t) is the deviation resulting from the difference between the set point r(t) and the controlled variable y(t), C(s) is the transfer function of the fractional corrector, $G_P(s)$ is the system transfer function.

Speed and position correction by fractional PID

The speed of the mobile robot depends on the speed of the wheels, i.e. the speed of the DC motor. The controller parameters depend on the relationship between the speeds and the desired set point.

To do this, we will take the following steps:

1) Mathematically model the DC motor
2) Write the relationship between the speed of the wheels and the speed of the robot
3) capture the speed and coordinates of the mobile robot
4) design the fractional order PID
5) Designing the algorithm
6) run the program on an Arduino UNO board

Relationship between wheel speed and robot speed

This wheel has its speed of rotation w_r about its axis. The longitudinal speed of the center of the wheel about the ground is denoted v. i.e. the speed at which the mobile move is v. The surfaces of the wheels are assumed to be smooth. So we have d=rθ

Deriving both members, we obtain $v=R\omega_r$.

By using speed reducers, the angular speed of the wheels is proportional to the speed of rotation of the DC motor $\omega_r = \frac{\omega}{k}$

So $$\frac{v(s)}{\omega(s)} = \frac{R}{k} \qquad (2)$$

With R is resistance and k is constant

For small wheels with a radius of 3 cm and a ratio reduction of 20, we have

$$\frac{v(s)}{\omega(s)} = \frac{0.03}{20} = 0{,}0015$$

Speed sensor

This sensor is shown in figure 2

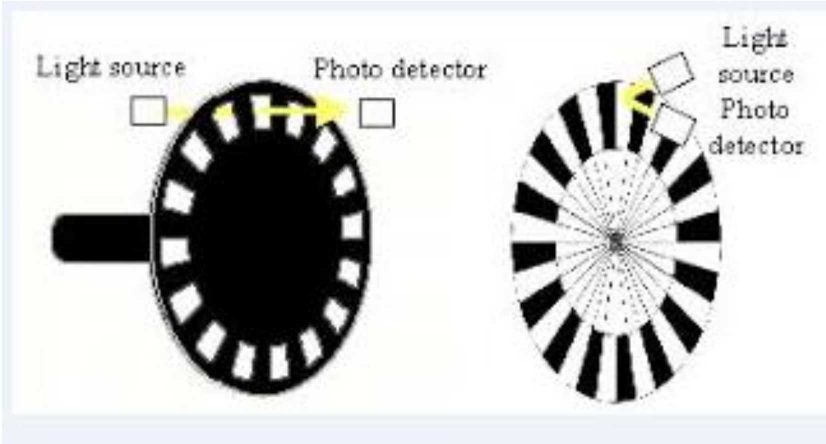

Figure 2:Encoder wheel

To detect speed, we use a speed sensor called an encoder wheel. The function of an encoder wheel is to send back a signal that depends on the movement made by the wheel. Industrial encoding wheels generally consist of a light beam and a presence of the beam and sends back an electrical signal. The faster the wheel rotates, the closer together the signal slots will be.

The speed of rotation of the motor is proportional to the frequency of the signal and the direction of rotation of the motor depends on the shape of the square-wave signal as shown in figure 3

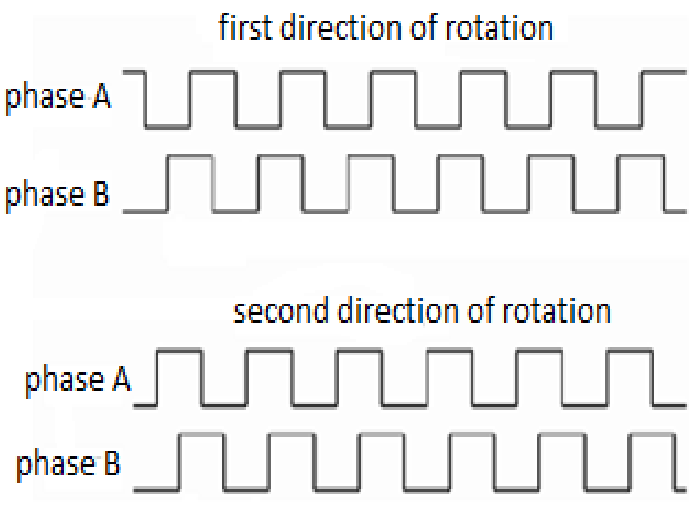

Figure3: Sensor output signal

PSO method

In particle swarm optimization (PSO), each individual in the population is referred to as a particle, while the population is known as a swarm. It should be noted that the particle can benefit from the movements of other particles in the same population to adjust its position and speed during the optimization process. Each individual uses the local information it can access about the movement of its nearest neighbors to decide on its movement.

Particles are described by their properties and have position and velocity characteristics.

The position of each particle represents a point in the search space, which is a possible solution to optimization problem, and the velocity is used to designate the direction to a new position. The properties of the particles change with each iteration.

They are updated by the equations:

$$\begin{cases} v_{ij}^{k+1} = wv_{ij}^{k} + c_1 \text{rand}\left(p_{bestij} - x_{ij}^{k}\right) + c_2 \text{rand}\left(g_{bestij} - x_{ij}^{k}\right) \\ x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1} \end{cases} \quad (3)$$

Or $i=1,2,3,\ldots,N_P$ et $j=1,2,3,\ldots,N_d$ et $k=1,2,3,4,\ldots,i_{termax}$

with :

N_P : is the number of particles in the swarm

N_d : is the number of variables in the problem (size of particle)

i_{termax} : is the maximum number of iterations

v_{ij}^{k} : is the velocity of the j^{rd} component of i^{rd} particle of the swarm at k^{rd} iteration.

c_1 and c_2 : are the ratio acceleration that characterize the particle's ability to search in another part of the search space, or to refine its search at the current location. In general, we choose c_1 and c_2 such that :$c_1+c_2=4$

: is random number between 0 and 1.

: is weighting that changes with each iteration. It is calculate by the expression:

$$w(\text{iter}) = w_{max} + \frac{w_{max} - w_{min}}{i_{termax}} \text{iter} \quad (4)$$

with:

: is the rank of the current iteration

w_{max} : is the initial value of the weighting generally taken to be 0.9

w_{min} : The final value of the weighting is between 0.3 and 0.4

PSO algorithm

The PSO algorithm consists of the following steps

- Initialize a population of particles and velocities, uniformly distributed in the search space D, and set stopping criterion
- Evaluate the objective function for each particle
- Update the best position p_{bestij} for each particle and the best overall position g_{bestij} in the population
- Update position and speed using previous equations
- Check the stopping criterion. If the maximum iteration number is not satisfied, go to step 2, otherwise, the program terminates, and the optimal solution is produced

Figure 8 below shows the flow chart that summarizes these steps

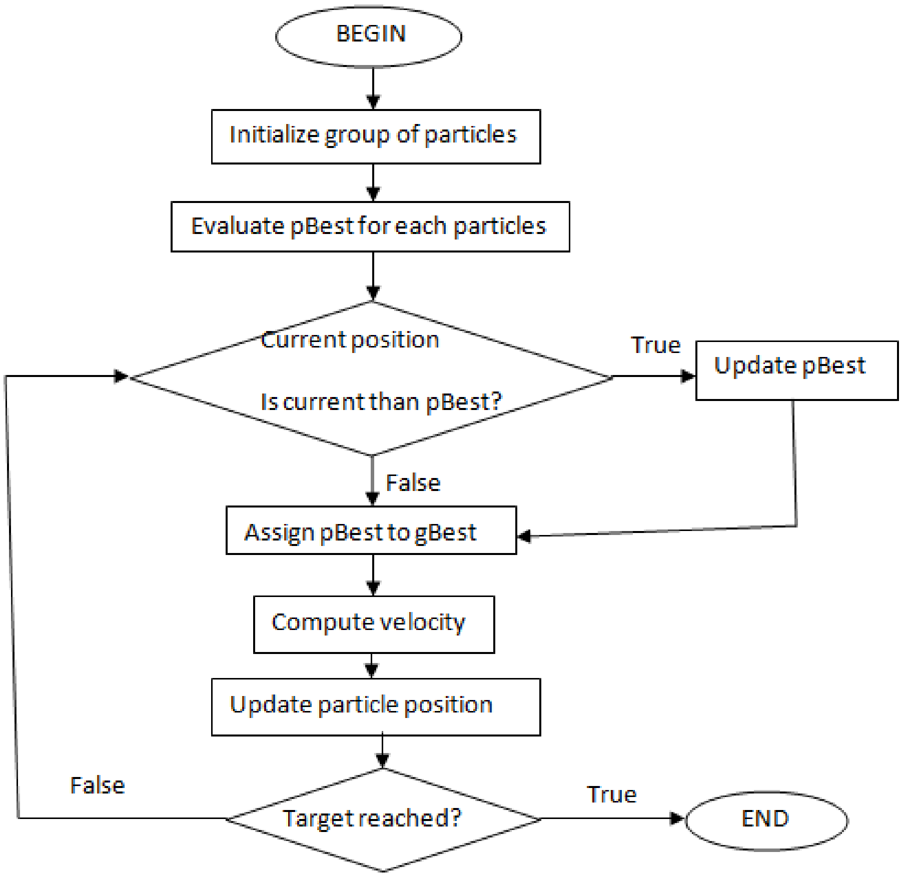

Figure 4: PSO organization chart

Application to a wheel robot

We will consider a vector containing the five parameters of the fractional corrector. These parameters will therefore be represented as particles or chromosomes. By following the steps of the algorithm and for a certain number of iterations, we have a high probability of finding a reliable solution to our optimization problem. The transient regime must be characterized by low overshoot and optimum response time to achieve good dynamic accuracy. To achieve this, the controller parameters are chosen to minimize the dynamic error **e(t) = y(t) -*yd*(t)**

When the algorithm is initialized, each chromosome contains five genes corresponding to the parameters of the fractional regulator $PI^{\lambda}D^{\mu}$.

Result and interpretation

The following table shows the speed values in steady state according to the instructions

Table 1 : speed and instruction:

System	Without regulation	With regulation
Speed instruction 1V	0.8 m/s	0,3 m/s
Speed instruction 4V	3,2 m/s	1,2 m/s

The following table shows the angles values in steady state according instructions

Table 2:angle and instruction

System	Without regulation	With regulation
Angle instruction 1V	The system is unstable	1,26 radian
Angle instruction 4V	The system is unstable	5,04 radian

The output values show that the fractional order PID controller corrects the error, and the output corresponds to the result according to the set point. Then, regulation also makes the system stable: the system without regulation is unstable.

Conclusion and Discussion

In conclusion, the FOPID can regulate all error variables: position, angle and speed. Next, we showed the actuators and sensors that enable the robots to pick up information from their surroundings. We have shown the formulas needed to control the robot, and now we need to know how the robot's physical variables change.

References

1. A. Dobra, General classification of robots. Size criteria, in Robotics in Alpe-Adria-Danube Region (RAAD), 2014 23rd International Conference on, 20 14, pp. 1-6.
2. B. Bayle, «Robotique Mobile »,Ecole Nationale Supérieure de Physique de Strasbourg, Université Louis Pasteur, 2007.
3. Eskimon, [Arduino 601] Le moteur à courant continu Posté le 12 décembre 2013, 12:08 http://eskimon.fr/285-arduino-601-le-moteur-courant-continu.
4. I. Podlubny, Fractional Differential Equations, San Diego, USA: Academic Press, 1999.
5. I. Podlubny, Fractional Order Systems and PIλDμ Controllers, IEEE Transactions on Automatic Control, vol. 44, no. 1, pp. 208-214, 1999.
6. Jean-Noël Montagné , Livret Arduino en français, Centre de Ressources Art Sensitif, November 2006.
7. J. Y azbeck, "Accrochage immatériel sûr et précis de véhicules automatiques," Université de Lorraine, 2014.
8. K.S. Miller, B. Ross, An Introduction to the Fractional Calculus and Fractional Differential Equations, New York, USA: Wiley, 1993.
9. Nicole Merle-Lamoot, Gilles Pannetier, Le développement industriel futur de la robotique personnelle et de service en France,2012.
10. Principe de l'accéléromètre piézo-électrique [archive], sur le site brouchier.com, consulté le 18 septembre 2013.
11. SALHI Farid, REZKI Dahmane,« Commande à distance d'un chariot mobile avec évitement d'obstacles ». Université Mouloud Mammeri de Tizi-Ouzou, 2010.
12. Y. Tang, M. Cui, C. Hua, L. Li, and Y. Yang, "Optimum Design of Fractional Order PIλDμ Controller for AVR System Using Chaotic Ant Swarm", Expert Systems with Applications, vol. 39, no. 8, pp. 6887-6896, 2012.