Genetic Algorithm Based Optimal Design of a PID Controller for trajectory tracking of a Mobile Robot

Prof Dr. Turki Y. Abdalla
Department of Computer Engineering
University of Basrah,
Basrah, Iraq

Seaar J. Al_Duboni
Department of Computer Engineering
University of Basrah,
Basrah, Iraq

Abstract

This paper deals with the modeling and control strategies of the motion of wheeled mobile robot. The model of the mobile robot has two driving wheels and the azimuth and velocity are dependently controlled by two PID controllers. The PID controller is one of the earliest and famous industrial controllers. It has many advantages: It is economic, simple easy to be tuned and robust. The tuning of these controllers is governed by system nonlinearities and continuous parameter variations. This paper deals with the optimal design of a PID controller for path tracking of mobile robot by using genetic algorithms (GA). The designed controller is tested for different paths.

Key words: Mobile Robot, Trajectory-Tracking Control, Simulink, PID Controller and Genetic Algorithms.

التصميم المثالي لجهاز سيطرة من نوع (PID) استنادا على الخوارزمية الوراثية لتتبع مسار إنسان آلي نقّال

أ.د.تركي يونس عبد الله
كلية الهندسة - قسم هندسة الحاسبات
جامعة البصرة - عراق

م.م. ستّار جواد الدبوني
كلية الهندسة - قسم هندسة الحاسبات
جامعة البصرة - عراق

الملخص

يتعاملُ البحث مع إستراتيجياتِ السيطرةِ وعرض حركةِ الإنسان الآلي النقّال ذو العجلات. نموذجُ الإنسان الآلي النقّال لَهُ عجلتا قيادة حيث يتم السيطرة على هذه العجلتين من خلال زاوية ميل الإنسان الآلي (Azimuth) و سرعته (Velocity) وذلك بواسطة جهازي سيطرة من نوع (PID). جهازَ سيطرة (PID) أحد أشهر أجهزةِ السيطرة الصناعية، حيث لَهُ العديد مِنْ الفوائد منها انه اقتصادي و بسيطٌ لكي يُستَمَّ (be Optimized) و كذلك متين. لتضبيط هذه الأجهزةِ مَحْكُومةُ بالنظامِ غير خطي (nonlinearity)، حيث تم استعمال الخوارزمية الوراثية (Genetic Algorithm) من أجل حصول على أفضل تصميم لمسيطر(PID) لتتبع طريقِ الإنسان الآلي النقّال، وتم اختبار متانة هذا المسيطر من خلال تتبع طرقٍ مختلفة.

1. Introduction

Autonomous robots may act instead of human beings. The robots are able to accomplish many tasks in dangerous places where humans cannot enter, such as sites where harmful gases or high temperature are present, a hard environment for humans. Cleaning robots and cargo delivery can work automatically and save costs by performing various routine tasks [1,2].

The PID controller has been used to control about 90% industrial processes worldwide [3]. The main problem of that simple controller is the correct choice of the PID gains. To tune the PID controller, there are number of strategies, the most famous, which is frequently used in industrial applications, is the Ziegler-Nichols method [4,5,6]. Moreover, GA was another method for tuning procedure. The advantage of tuning with GA is the ability of choosing controller gains which optimize drive performance based on multi objective criterion without tripping in a local minima solution [4].

2. PID and GA

Despite the development of more advanced control strategies, the majority of industrial control systems still use PID controllers because they are standard industrial components, and their principle is well understood by engineers, which is most widely used control structure in the control industry, this is because PID controllers are easy to tune, easy to implement, and available at little cost [3].

It can give a good performance for stable linear processes. Self-tuning and adaptive PID design approaches can overcome the operating point varying parameters [5].

The controller output U in S-domain is given by the following equation:

$$U(s) = K\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + (T_d s / N)}\right)E(s) \quad \ldots (1)$$

Where:

Error (E) = reference (r) - actual (y).

K: Proportional gain.

Ti: Integral time.

Td: Derivative time.

N: Filter factor to limit the noise generated in the derivative action [5,3,7]. Fig.1 shows the structure of PID.

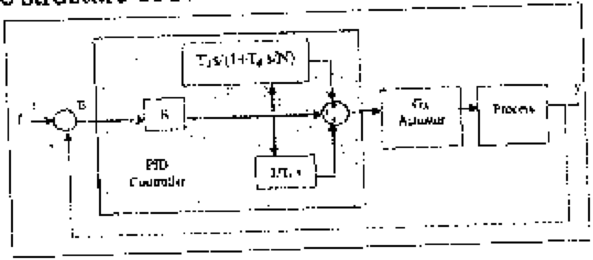


Fig.1 General Structure of PID control loop

A proportional controller (K_p) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error. An integral control (K_i) will have the effect of eliminating the steady-state error, but it may make the transient response worse. A derivative control (K_d) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response [8].Note that these correlations may not be exactly accurate, because Kp, Ki, and Kd are dependent of each other. In fact, changing one of these variables can change the effect of the other two.

In this paper, we propose a GA-based optimal design of a PID controller to solve the control problem of a mobile robot. The application of GAs can broadly be classified into two distinct areas: off-line optimization and on-line optimization.

Off-line optimization have proved to be the most popular and successful.

On-line optimization tend to be quite rare because of the difficulties associated with using a GA in real-time and directly influencing the operation of the system [9]. It operates on the principle of the survival of the fittest. A constant size population of individuals, each of them is represented by a fixed number of parameters which are coded in binary form (chromosomes); encode possible solutions of a given problem. An initial population of individuals (possible solutions) is generated at random by using *trial and error* method to obtain a good result. The allowable range of variation for each parameter is given. There are three main operators that constitute the genetic algorithm search mechanism: selection, crossover and mutation. In every evolutionary step, known as a generation, the individual of the current population (or family) are decoded and evaluated [10,11].

The three operators are implemented iteratively, each iteration produces a new population of solutions (generation). The genetic algorithm continues to apply the operators and evolve generations of solutions until a near-optimum solution is found or the maximum number of possible generations is produced. Fig.2 shows the algorithm flowchart. From simulation results, the near-zero errors for Azimuth and Velocity can be achieved with appropriate controller parameters tuning based on GAs.

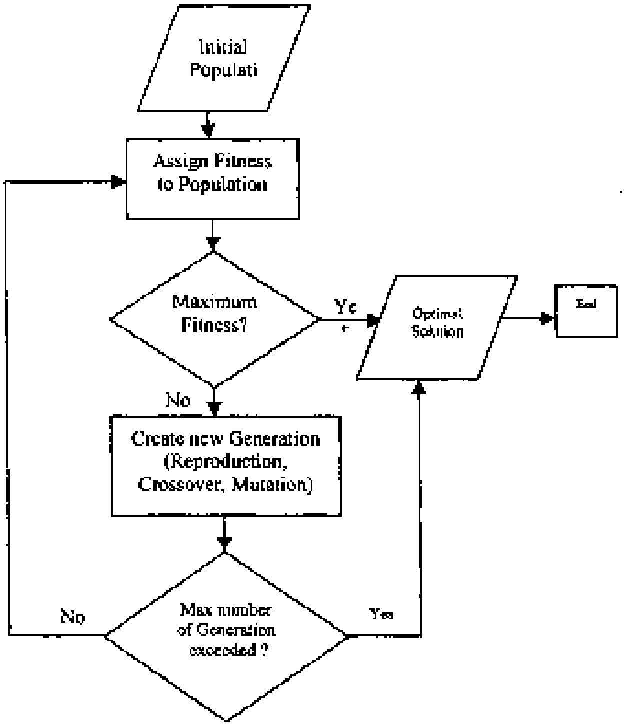


Fig.2 The genetic algorithm flow chart

3. Dynamic Model

Mobile robots have received a great deal of research in recent years. A significant amount of research has been published in many aspects related to mobile robots. Most of the research is devoted to design and develop some control techniques for robot motion and path planning [12]. A large number of researchers such as [13-15] have used kinematic models to develop motion control strategy for mobile robots. Their argument and assumption that these models are valid if the robot has low speed, low acceleration and light load [12].

However, dynamic modelling of mobile robots is very important as they are designed to travel at higher speed and perform heavy duty work. This paper uses dynamic model and control strategy for wheeled mobile robot on two wheels and a castor. The mobile robot considered here is shown in Fig.3. It consists of a vehicle with two driving wheels mounted on the same axis, and a front passive wheel for balance. The two driving wheels are controlled independently by motors. Let the mobile robot be rigid moving on the plane.

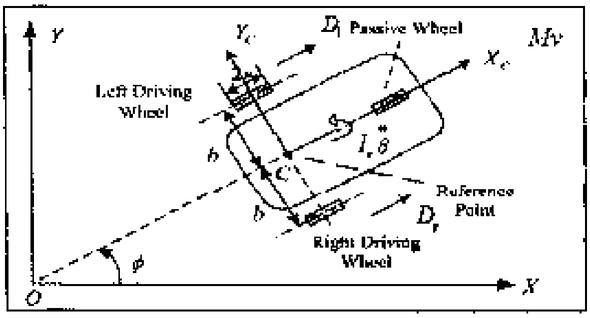


Fig.3: Model of mobile robot

We assume that the absolute coordinate system OXY is fixed on the plane. Then, the dynamic property of the robot is given by the following equation of motion [16]:

$$I_v \ddot{\theta} = D_r l - D_l l \qquad \dots(2)$$

$$M \dot{v} = D_r + D_l \qquad \dots(3)$$

For the right- and left-wheels, the dynamic property of the driving system becomes

$$I_w \ddot{\phi}_i + c\dot{\phi}_i = ku_i - rD_i \qquad \dots(4)$$

$$(i = r, l)$$

Where each parameter and variable is defined as follows:

I_v: moment of inertia around the C.G. of robot

M : mass of the robot

D_l, D_r: left and right driving forces

l : distance between left and right wheel and the c.g. of robot

θ: azimuth of the robot

v : velocity of the robot

I_w: moment of inertia of wheel

c : viscous friction factor.

k : driving gain factor.

r : radius of wheel.

ϕ_i: rotational angle of wheel.

u_i : driving input.

On the other hand, the geometrical relationships among variables θ, v and ϕ_i are given by

$$r\dot{\phi}_r = v + l\dot{\theta} \qquad \dots(5)$$

$$r\dot{\phi}_l = v - l\dot{\theta} \qquad \dots(6)$$

From these equations, defining the state variable for the robot as

$$x = \begin{bmatrix} v & \theta & \dot{\theta} \end{bmatrix}^T$$

The manipulated variable as $u = [u_r, u_l]^T$, and the output variable as $y = [v \quad \theta]^T$ yields the following state equation:

$$\dot{x} = Ax + Bu \qquad \dots(7)$$

$$y = Cx \qquad \dots(8)$$

$$A = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & a_2 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 & b_1 \\ 0 & 0 \\ b_2 & -b_2 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$ Where

$$a_1 = -\frac{2c}{Mr^2 + 2I_w}, \quad a_2 = -\frac{2cl^2}{I_v r^2 + 2I_w l^2}$$

$$b_1 = \frac{kr}{Mr^2 + 2I_w}, \quad b_2 = -\frac{krl}{I_v r^2 + 2I_w l^2}$$

The mobile robot physical parameters are given in Table1:

Table1: The values of known physical parameters of the mobile robot

Parameter	Value	Unit
I_v	10	$Kg.m^2$
M	200	Kg
l	0.3	m
I_w	0.005	$Kg.m^2$
c	0.05	Kg/s
r	0.1	m
k	5	-

To simulate the above mobile robot model, we use the State-Space block in Matlab simulation environment as shown in Fig.4 with 4th order Runge-Kutta-Gill method with an integration step 1[ms]. The velocity and azimuth of the robot are controlled by manipulating the torques for the left wheels (U_l) and the right wheels (U_r).

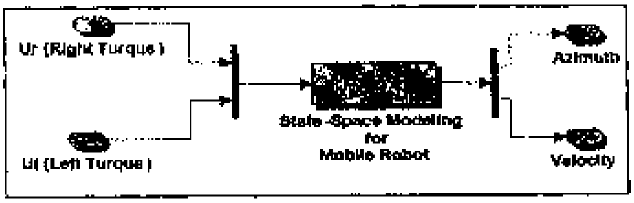


Fig.4: Block diagram for the mobile robot model in Matlab/Simulink environment

4. GA-Based PID Controller Tuning

Genetic algorithms have been utilized in robotics for both path planning and the design of behavioral controllers [9]. As a mathematical means for optimization, GAs can naturally be applied to the optimal-tuning of PID controllers. With reference to a step input signal for velocity and deviation line with slope (-5) for azimuth, the role of the PID controller is to drive the output response within the user's specifications. Obviously, the parameter settings of the PID controller should be fine-tuned so as to meet as high requirements as possible. Optimization of PID controllers firstly needs design the optimization goal, and then encode the parameters to be searched. Genetic operator is running until the stop condition is satisfied. The decoding values of the last chromosome are optimized parameters of the PID controller. To obtain the optimal controller, the following implementations of the genetic algorithm are used.

Parent 1	a	b	c	d	e	f	g	h
Parent 2	1	2	3	4	5	6	7	8

Child 1	a	b	3	4	e	6	7	8

5. Representation of Parameters

For most applications of genetic algorithms to optimize problems, the real coding technique is used to represent a solution to a given problem [17]. In this work, real valued representation is used. This is for many reasons; the first is that the values that deal with are real, then to prevent encoding of the floating point values with a binary encoding and this need more genes, the second, is the precision, the third; are the processes of encoding and decoding take much time. Each chromosome represents a solution of the problem, it consists of eight genes: : vector = $[K_{pr}\ K_{dr}\ K_{ir}\ K_{nr}\ K_{pl}\ K_{dl}\ K_{il}\ K_{nl}]$, it represents (, and) for PID_Azimuth controller gains (right torque) with right scaling gain ( ) and (, and) for PID_Velocity controller gains (left torque) with left scaling gain ( ). It must be noted here that the searching area of each gain (gene) must be specified (0 to 100) with 10 orders precision for floating point value.

i.Crossover

Crossover allows an improvement in the species in terms of evolution of new solutions at random on each parent and then , complementary fractions from the two parents are linked together to form a new chromosome[10,11].

There are several functions of crossover (Scattered, Single point, Two point, Intermediate, Heuristic Arithmetic and line) [18], by experience for each type to obtain the minimum fitness, the scattered crossover is best, so, the scattered crossover will be used here.

This is to create a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child [19]. For example, if p1 and p2 are the parents.

And the binary vector is **[1 1 0 0 1 0 0 0]**, the function returns the following child:

ii. Mutation

The mutation operator alters a copy of a chromosome reintroducing values that might have been lost or creating totally new features [10,11].

There are several functions of mutation (Gaussian , Uniform and Adaptive Feasible), by experience for each type to obtain the minimum fitness, the Adaptive Feasible mutation is best, so, the Adaptive Feasible mutation will be used in this paper. Which is randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation.

The feasible region is bounded by the constraints and inequality constraints. A step length is chosen along each direction so that linear constraints and bounds are satisfied [19].

iii. Fitness Function

Fitness function decides whether a chromosome will contribute to the next generation [10].

To evaluate the controller performance, the Mean Square Error (***MSE***) is used as a fitness function [20]:

$$I_{MSE} = \frac{1}{n}\sum_{K=1}^{n}(e(K))^2 \qquad \dots (9)$$

Where ***n*** represents number of samples, ***e(t)*** is the error for each Azimuth and Velocity as following in the equations.

$$MSE\_Azimuth = \frac{1}{n}\sum_{K=1}^{n}(\phi_d(K) - \phi(K))^2$$

$$MSE\_Velocity = \frac{1}{n}\sum_{K=1}^{n}(v_d(K) - v(K))^2$$

$$MSE\_Total = MSE\_Azimuth + MSE\_Velocity$$

Where v_d, φ_d are the desired velocity and the desired azimuth, respectively. v, φ are the actual velocity and the actual azimuth of the robot, respectively.

Minimization of this MSE ensures that the system reaches its final state quickly as well as steady state error is small, in this paper, the *MSE_Total* will be used as fitness function.

iv.Select Probability

The selection procedure depends on the value of the fitness function. Individuals with low-fitness have a better chance of reproducing, while high-fitness ones will disappear [10,11].

To maintain a reasonable differential between relative fitness ratings of chromosomes and to prevent a too-rapid takeover by some super chromosomes, there are several functions for selection (Stochastic Universal Sampling, Remainder, Uniform, Roulette and Tournament), by experience for each type to obtain the minimum fitness, the Stochastic Universal Sampling selection is best, so, the Stochastic uniform selection will use in this paper.

The stochastic uniform, lays out a line in which each parent corresponds to a section of the line of length proportional to its scaled value.

The algorithm moves along the line in steps of equal size.

At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size [18,19].

v. Termination Condition

Maximum generation termination method is used to decide whether the termination condition is satisfied or not.

6. Simulations results

The block diagram of the mobile robot control system using Matlab/Simulink environment for the closed loop system model with two PIDs controllers is shown in Fig.5, one PID controller used for control of velocity and another for control azimuth of mobile robot. The velocity and azimuth of the robot are controlled by manipulating the torques for the left- and the right-wheels.

The velocity error e_v and the azimuth error e_θ are considered as the inputs, and the driving torque required for controlling the two wheels u_l and u_r are considered as the output. Here, the input deviations e_v and e_θ are defined by

$$e_v = v_d - v \qquad \dots (10)$$
$$e_\theta = \theta_d - \theta$$

Where v_d , θ_d : are the desire velocity and the desire azimuth, respectively. v, θ: are the actual velocity and the actual azimuth of the robot, respectively.

The summaries of genetic algorithm parameters chosen for the tuning purpose of PID controller gains are shown in Table 2.

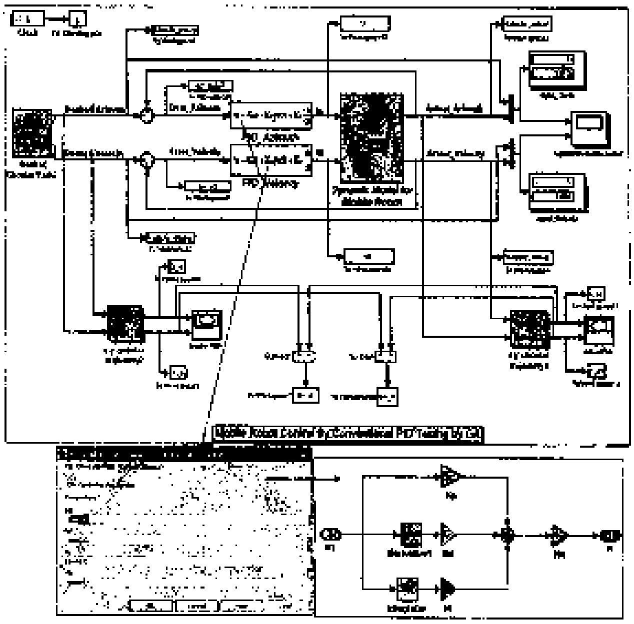


Fig.5: Simulation model of system control using two PID controllers in Matlab/Simulink environment

To simulate the mobile robot model, used the reference velocity v_d given as 1 [m/sec] and the reference azimuth to create circular trajectory is represented by the following equation:

$$\theta = (2\times\pi)/(m)\times f(t)[rad] \qquad \dots(11)$$

Where m (slope) = -5, f(t)=t { t=0, 1, 2 …5 sec}, the simulation of reference azimuth is carried out by using simulink blocks as shown in Fig.6.

Fig.6: Block diagram for desired azimuth

Table2: GA Parameters

No.	Parameters	Description
1	Chromosome encoding	Real encoding
2	Number of generations	500
3	No. of chromosomes in each generation	20 (19 randomly and 1 by Trial and Error)
4	No. of variables in each chromosome	8
5	T&E initial chromosome [KpR KdR KiR KuR KpL KdL KiL KuL]	[20,2.8,0.2,5,32,2.9,1.33,0.2]
6	Fitness function	Mean Squire Error (MSE)
7	Selection method	Stochastic Universal Sampling
8	Crossover method	Scattered
9	Mutation method	Adaptive Feasible
10	Termination	Number of generation

When the GAs parameters in table2 are used by Matlab optimization tool of GA, it displays a plot of the best and mean values of the fitness function at each generation. When the algorithm stops, the plot appears as shown in Fig.7, the points at the bottom of the plot denote the best fitness values (dote points), while the points above them denote the averages of the fitness values in each generation (star points). The plot also displays the best and mean values in the current generation numerically at the top.

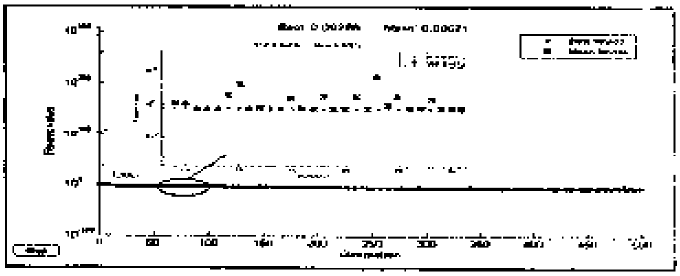
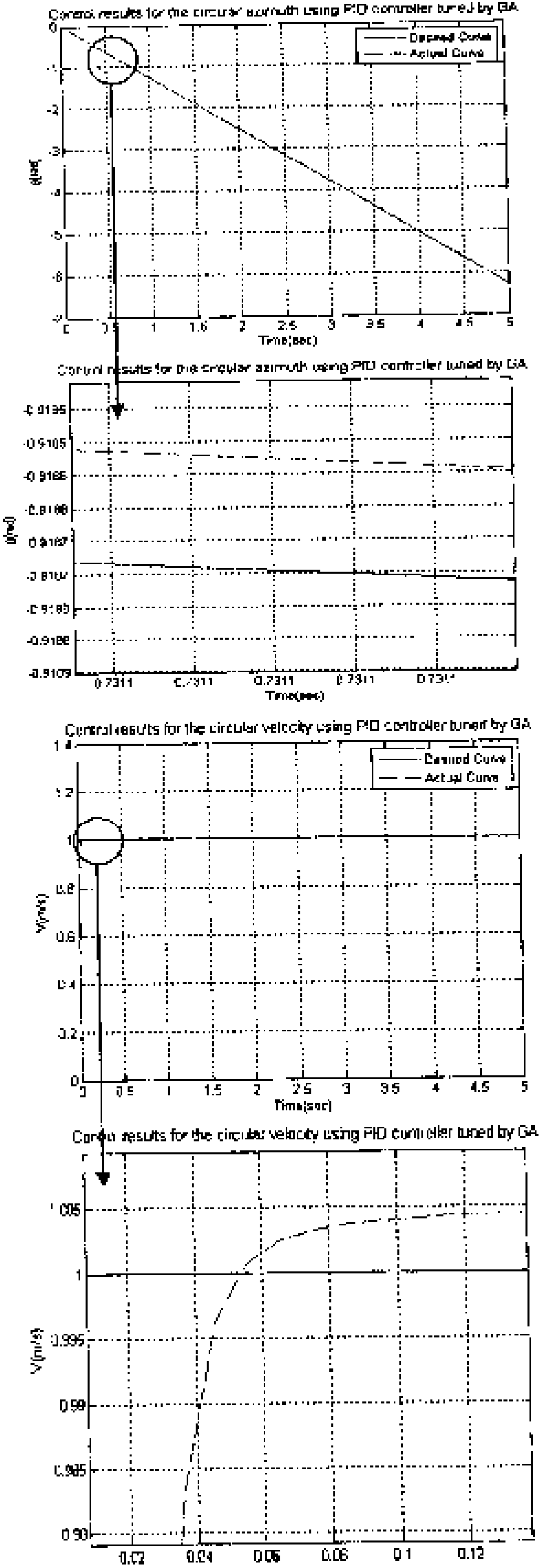


Fig.7: MSE vs. generations for PID controller tuned by GA

Fig.(8) illustrates the response for azimuth, error of azimuth, velocity, and error of velocity for PID controller tuned by Genetic algorithm.



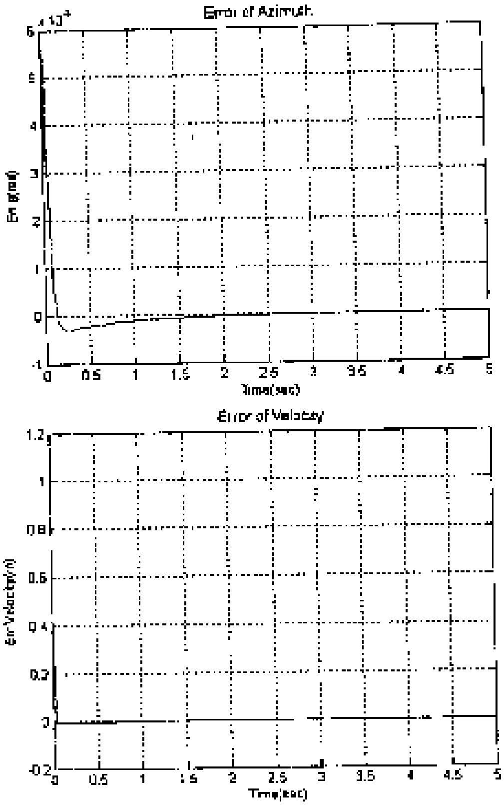


Fig.8: Azimuth, Velocity and Errors for each one for using PID controller tuned by GA

In order to transfer the azimuth (θ) and velocity (v) of mobile robot into the trajectory, we use the following equations [21]:

$$\dot{x} = v\cos(\theta) \qquad \dots (12)$$

$$\dot{y} = v\sin(\theta)$$

The x-y transducer block transfer Velocity and Azimuth into (Xc, Yc) coordinates in Matlab / simulink is illustrated as shown in Fig.9, Fig.10 shows the circular trajectory when applied the x-y transducer block.

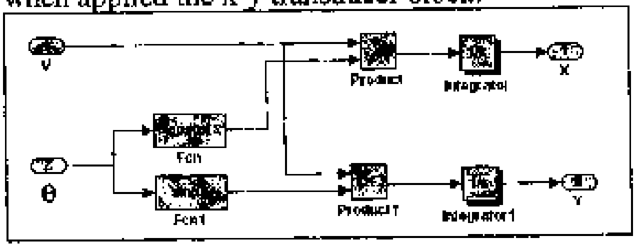


Fig.9: A simulink model of transducer blocks to transfer from velocity and azimuth into (Xc, Yc) coordinates

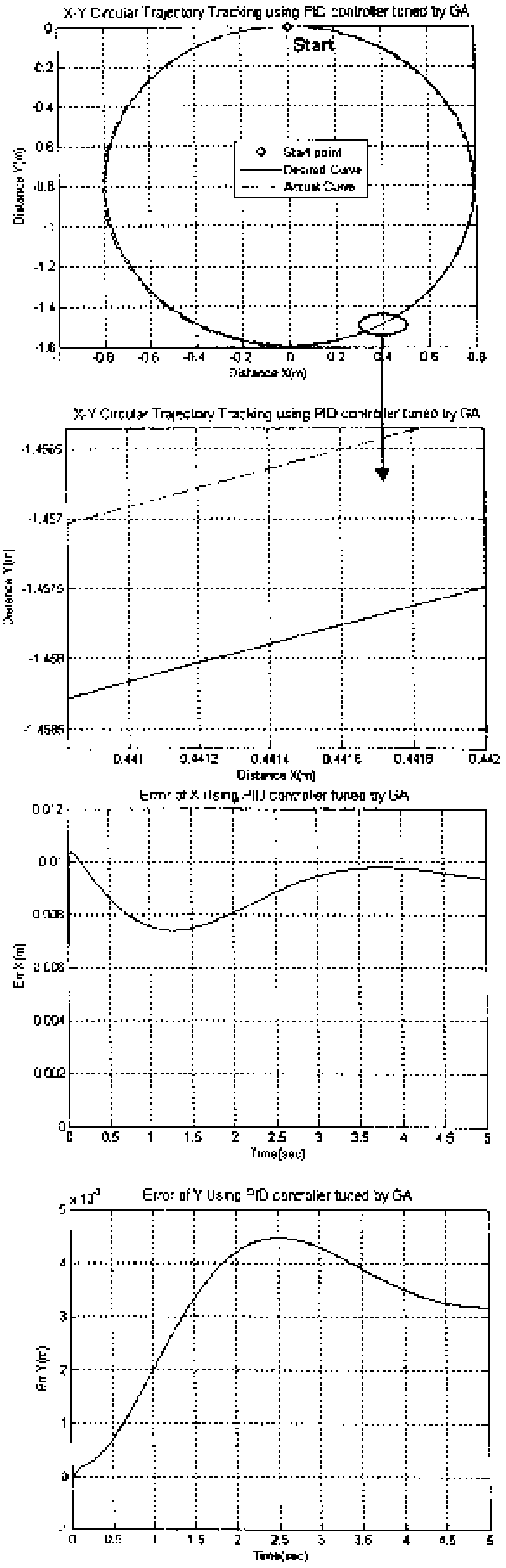


Fig.10: Circular trajectory and x-y errors when used PID controller tuned by GA

Table 3 illustrates the Mean Square Error for each the initial gains (first iteration) ,which is obtained by using trial and error (T&E) tuning method, and the finial iteration of gains by using genetic algorithm method, this table has the percentage improvement between them (T&E and GA), the percentage improvement represented by the following equation:

$$((Old\_value - New\_value)/(Old\_value))\times 100 \quad (13)$$

Table3: Mean Square Error's (MSE's) values for PID controller tuned by GA with percentage improvement

Gains by trail and error (initial gains)		
gain=[20,2.8,0.2,5,32,2.9,1.33,0.2]		
MSE_Azimuth	**MSE_Velocity**	**MSE_Total**
8.08E-05	**5.74E-02**	**0.0575**
Gins by GA (finial iteration of GA)		
gain=[79.5000 2.8000 95.7000 65.0000 96.6250 0.5250 47.3300 4.9453]		
MSE_Azimuth	**MSE_Velocity**	**MSE_Total**
2.30E-07	**4.80E-03**	**0.0048**
Percentage Improvement between trial and error and GA		
MSE_Azimuth%	**MSE_Velocity%**	**MSE_Total%**
99.71%	**91.64%**	**91.65%**

In order to test the robustness of PID controller, applied this controller for tracking several different paths as follows:

1) **Circular trajectory tracking**
2) **Line trajectory tracking**
3) **Wave trajectory tracking**

The test is done with and without loading.

a) **Without loading**

1) **Circular trajectory tracking**: the reference velocity is given as 1 [m/sec] and the reference azimuth is represented by Eq (11).
2) **Line trajectory tracking**: the reference velocity is given as 4.243 [m/sec] and the reference azimuth is 0.7854 [rad] (45°) to arrive a target point (15, 15)) in Fig.11.
3) **Wave trajectory tracking**: the reference Velocity and Azimuth is sine wave with amplitude = -1 to 1 and frequency = 3 [rad/sec] as shown in Fig.12.

Table4 illustrates the MSE for wave and line tracking.

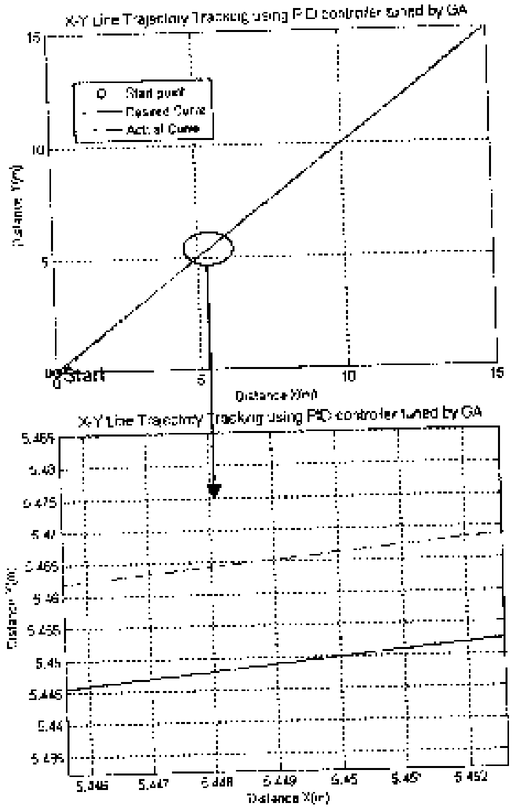


Fig.11: Line Trajectory Tracking

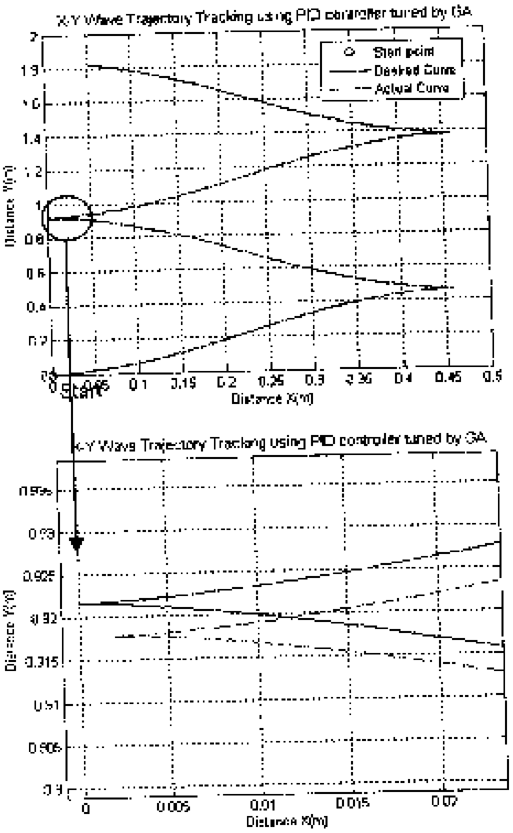


Fig.12: Wave Trajectory Tracking

b) With Load

The disturbances will be represented by another mass added to the initial mass (M0) as follows:

$M1= (M0+0.1\ M0) = (200+20) =220$ Kg
$M2= (M0+0.2\ M0) = (200+40) =240$ Kg
$M3= (M0+0.5\ M0) = (200+100) =300$ Kg

The mass of the mobile robot is changed from 200 Kg to 220,240 and 300Kg, the simulation is performed for each all previous paths. Table5 illustrates the MSE for all paths with different loading.

Table4: MSE's values for different paths

No.	Approach	MSE_Azimuth	MSE_Velocity	MSE_Total
1	Line Path	0.0111	0.0184	0.0295
2	Wave Path	0.0011 e-003	0.2269e-003	0.228e-003

Table5: MSE's values for different paths and loading

Mass(M1)=220 Kg

	Approach	MSE_Azimuth	MSE_Velocity	MSE_Total
1	Circular Path	3.5729e-007	0.0063	0.0063
2	Line Path	0.0109	0.0215	0.0324
3	Wave Path	0.0013e-003	0.2332e-003	0.234e-003

Mass(M1)=240 Kg

No.	Approach	MSE_Azimuth	MSE_Velocity	MSE_Total
1	Circular Path	4.1695e-007	0.0071	0.0071
2	Line Path	0.0103	0.0279	0.0382
3	Wave Path	0.0015e-003	0.2589e-003	0.260e-003

Mass(M1)=300 Kg

No.	Approach	MSE_Azimuth	MSE_Velocity	MSE_Total
1	Circular Path	6.3739e-007	0.0062	0.0062
2	Line Path	0.0085	0.0268	0.0353
3	Wave Path	0.0018e-003	0.2700e-003	0.271e-003

7. Conclusions

The paper deals with the modeling and control strategies of the motion of wheeled mobile robots. The model of the vehicle has two driving wheels and the angular velocities of the two wheels are independently controlled. This paper has suggested a fine-tuning technology for optimal design of a PID controller for a mobile robot based on genetic algorithms. The designed controller is tested for different paths. Simulation results show good performance for the designed control structure. Also it shows good robustness.

References

[1] Y. Shinoda, "Collaborative, **"Motion Planning of Autonomous Robots"**, M.Sc. Thesis, School of Information Science, Advanced Institute of Science and Technology, Ishikawa Japan, 2007.

[2] R. A. Felder, **"Mobile Robot Simulation of Clinical Laboratory Deliveries"**, M.Sc. Thesis, the University of Virginia, U.S.A, 1998.

[3] A. Haj-Ali and H. Ying , **"Structural Analysis of Fuzzy Controllers with Nonlinear Input Fuzzy Sets in Relation to Nonlinear PID Control with Variable Gains"**, Associate Editor Gary, Wayne State University , Detroit , MI48202 ,USA ,2004, available on the following link: www.sciencedirect.com

[4] S. M. Gadoue, D. G. and J. W. Finch, **"Tuning of PI Speed Controller in DTC of Induction Motor Based on Genetic Algorithms and Fuzzy Logic Schemes"**, International Journal of Advanced Robotic Systems, University of Newcastle upon Tyne, pp.187 - 194, ISSN 1229-3406, 2005.

[5] W. Gharieb, G. Nagib, **"Fuzzy Intervention in PID Controller Design"**, IEEE International Symposium on Industrial Electronics, Pusan KOREA, June, 2001.

[6] J. Jantzen, **"Tuning of Fuzzy PID Controllers"**, Technical University of Tech. Department of Automation, report no 98 - H871, Denmark, Sep, 1998.

[7] A. N. I. Wardana, **"PID-Fuzzy Controller for Grate Cooler in Cement Plant"**, Indonesia Cement and Concrete Institute Jalan Raya Ciangsana, Bogor, 16969, Indonesia, 1997.

[8] J. A. Shaw, **"The PID Control Algorithm, How it works, how to tune it, and how In Proc."**, IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2003.

[9] P. J. Fleming and R. C. Purshouse, **"Genetic Algorithms in Control Systems Engineering"**, University of Sheffield, Research Report No. 789, , UK, May, 2001.

[10] H. C. Chen and S. H. Chang, **"Genetic Algorithms Based Optimization Design of a PID Controller for an Active Magnetic Bearing"**, International Journal of Computer Science and Network Security, VOL.6 No.12, December, 2006.

[11] F. Xhafa, **"An Experimental Study on GA Replacement Operators for Scheduling on Grids"**, Technical University of Catalonia, pp: 927 – 932, Barcelona, Spain, 2004.

[12] A. Albagul1 and Wahyudi, **"Dynamic Modelling and Adaptive Traction Control for Mobile Robots"**, International Journal of Advanced Robotic Systems, Volume 1, Number 3 , ISSN 1729-8806, pp. 149 - 154,. Albagul, 2004.

[13] X. Jiang and Y. M. Xingquan **"Predictive Fuzzy Logic Controller for Trajectory Tracking of a Mobile Robot"**, IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications Helsinki University of Technology, Espoo, Finland, June, 2005.

[14] S. E. Mahmoudi , A. A. Bitaghsir , B. Forouzandeh and A. R. Marandi , **" A New**

Genetic Method for Mobile Robot Navigation", Electrical and Computer Engineering Dep. University of Tehran, P.O Box : 14395/515, IRAN, September, 2004.

[15] V. Mohan Peri and D. Simon" **Fuzzy Logic Control for an Autonomous Robot**" M.Sc. Thesis, Cleveland State University Cleveland ,Ohio ,USA, 2002.

[16] J. Peng, Y. Wang and W. Sun, "**Trajectory-Tracking Control for Mobile Robot Using Recurrent Fuzzy Cerebellar Model Articulation Controller**", Neural Information Processing – Letters and Reviews, Vol. 11, No. 1, January 2007.

[17] H. C. Chen and S. H. Chang, "**Genetic Algorithms Based Optimization Design of a PID Controller for an Active Magnetic Bearing**", International Journal of Computer Science and Network Security, VOL.6, No.12, December 2006.

[18] F. Peter, P. Hartmut and F. Carlos, "**GEATbx Introduction Evolutionary Algorithms: Overview, Methods and Operators**", M.Sc. Thesis, University of Sheffield , November, 2005.
http://www.geatbx.com

[19] **Genetic Algorithm Toolbox for Matlab**, Help file for Matlab, version (7.6.0) R2008a.
http://www.matlabworks.com

[20] G. Baker and S. Bin, "**The PID Controller Design Using Genetic Algorithm**", M.Sc. Thesis, University of Southern Queensland, October, 2005.

[21] G. Mester, "**Obstacle Avoidance of Mobile Robots in Unknown Environments**", International Symposium on Intelligent Systems and Informatics , pp:128 – 139, Subotica, Serbia, August, 2007.