Ant Colony Optimization Algorithm Based on Optimal PID Parameters for a Robotic Arm

Sid Ahmed Dahmane, Abdelwahab Azzedine, Abdelkader Megueni

Laboratory of Structures and Solids Mechanics (LMSS), University Djilali Liabès of Sidi Bel Abbès, (22000), Algeria

Abstract: - In this paper, the ACO (Ant Colony Optimization) algorithm is implemented as an adjustment mechanism for the PID controller. The objective of this hybrid controller is to improve the performance of a servo control of a loopback system or closed process. The search technique using the ACO algorithm is applied to find for the best gain parameters of the PID controller. The setting of a PID consists in obtaining an adequate response of the process. The objectives are robustness, speed and precision. One of the applications is the control of a manipulator arm using a DC servomotor as actuator. This study introduces the DC Servo PID controller design using SISTOOL PID Auto Tuning for high order models and implemented on an Arduino model Mega 2560 controller using the MATLAB / Simulink interface package. Better controller design can be achieved by utilizing PID optimization by the Ant Colony Method (ACO).The results of the simulation show the performance and efficiency of the use of the ACO algorithm for PID adjustment.

Key-Words: - PID corrector, ACO-PID control, Sisotool**,** DC Servo Motor, Arduino Mega 2560

I Introduction

PID controllers are frequently used in the control process to regulate the time domain behavior of many types of dynamic installations. These controllers are extremely popular because of their simple structure and they can usually provide a good closed-loop response characteristic. Despite its simple structure, it seems so difficult to find an appropriate PID controller [1]. Given this problem, different methods have been proposed to adjust these parameters.

As a result, many optimization methods are developed to adjust PID controllers, such as fuzzy logic [2, 3], neural network [4], fuzzy neural logic [5], immune algorithm [6] simulated annealing [7] and pattern recognition [8]. In addition, we have many other optimal PID optimization methods based on many random search methods, such as the genetic algorithm (GA) [9, 10], particle swarm optimization [11] and ant colony optimization [12].

In this work, we have developed a method for the problem of design PID controllers as an optimization approach taking into account the ant colony optimization (ACO) algorithm. The researchers pointed to the ACO's ability to efficiently search for and locate an optimal solution. This method was mainly inspired by the fact that

ants are able to find the shortest way between their nest and a food source.

Ant colony optimization (ACO) [13, 14] is a metaheuristic approach recently developed to solve hard combinatorial optimization problems such as the problem of the traveling salesman TSP [15], the problem of quadratic assignment [16], color problems [17], hydroelectric generation planning problems [18].

Ant colony optimization algorithms are particularly suitable for finding solutions to difficult optimization problems. A colony of artificial ants cooperates to find good solutions, which are an emerging property of the cooperative interaction of ants. Based on their similarities to ant colonies in the wild, ant algorithms are adaptive and robust and can be applied to different versions of the same problem as well as different optimization problems.

An ant colony of finite size collectively seeks a good solution to a given optimization problem. Each ant can find a solution or at least part of the solution to the optimization problem (but only when several ants work together can they find the optimal solution) Since the optimal solution can be found only through the global cooperation of all the ants of a colony, it is emerging result of this cooperation. In seeking a solution, ants do not communicate directly but indirectly by adding pheromone to the environment. On the basis of the specific problem, an ant is assigned a starting state and crosses a sequence of neighboring states seeking the shortest path.

Ants use this private and public information to decide when and where to deposit pheromones. In most applications the amount of pheromone deposited is proportional to the quality of movement performed by an ant. Thus, the more rich the pheromone, the better the solution found. Once an ant has found a solution it dies, that is, it is removed from the system.

This paper has two main contributions. First, a PID corrector was designed for the higher order system using Automatic PID Adjustment using the Matlab SISO Design Tool.

Secondly, for the same system, an ACO-PID corrector has been proposed. It has been established that the latter gives a better result from the point of view of the analysis of the performance (precision and response time).

The article is organized as follows, following the introduction (section I), section II describes the model of the manipulator - actuator arm assembly. Section III establishes the value of the transfer function of the DC servomotor. Section IV describes the design of the PID regulator for DC servo motors. Section V presents the ACO-PID design using the ant colony algorithm under MATLAB / Simulink. Section VI concludes this article.

II. ROBOT ARM MANIPULATOR

In order to demonstrate the effectiveness of the ACO-PID controller for accurate tracking, we will verify the controller by simulation on the arm manipulator model shown in Fig. 1. This model represents the robot arm manipulator with five degree of freedom. It was designed as one of the experimental platforms for research and education in our laboratory.

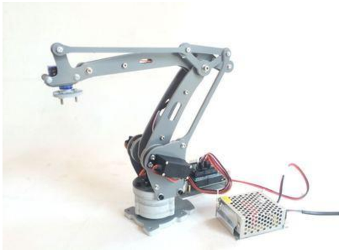

Fig. 1. Arm robot manipulator model

Furthermore, the main component of the robot arm manipulator model uses Arduino Mega 2560 in which the schematic of electronic hardware is shown in Fig. 2.

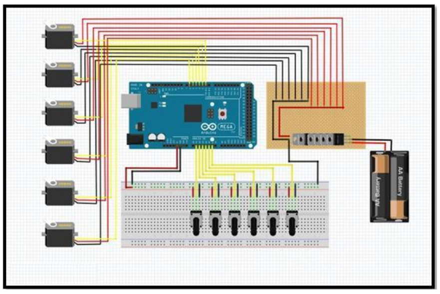

Fig. 2. Schematic of electronic hardware [17]

III. DC SERVO MOTOR DESCRIPTION

DC servo motors are most suitable for wide range of speed control and for many adjustable speed drives. For this simulation, Fig. 3 represents the DC servo motor model.

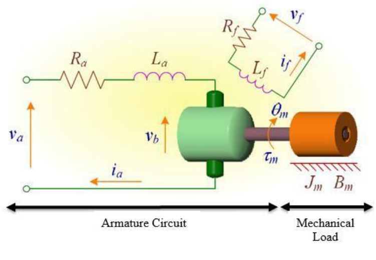

Fig.3. Schematic representation of the DC servo motor [4]

The purpose of modeling DC servo motor is to approach the actual DC servo motor [4]. By including the parameters we can get the transfer function of DC servo motor for controlling position [1].

$$G_{position}(s) = \frac{\theta(s)}{V_a(s)} = \frac{1.112s + 4.517}{s^2 + 4.187s + 6.265} \quad (1)$$

The block diagram of DC servo motor is shown in Fig. 4 with zero value of load torque (TL) [6].

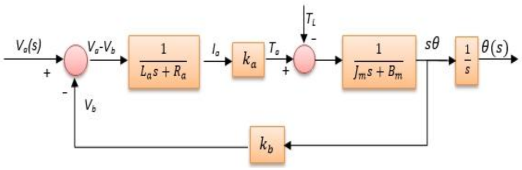

Fig. 4. Block diagram of DC servo motor.

IV. DESIGN OF THE PID CONTROLLER

A PID controller is being designed for a higher order system with transfer function in equation (1). Before starting the simulation and implementation of PID controller, we construct the design of PID controller using Matlab's SISO design tool and analyze the stability respond systems by means of the root locus method. The design architecture of Matlab's SISO design tool for automated PID tuning is shown in Fig. 5.

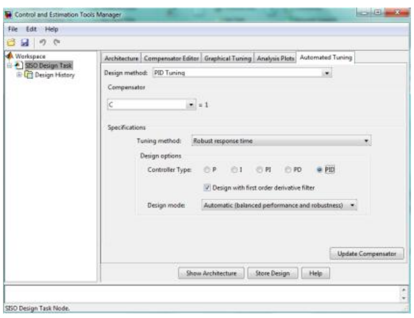

Fig.5. Automated PID tuning using Matlab's SISO design tool.

Using automated PID tuning, we can get the root locus stability for the open loop system G(s) as shown in Fig. 6.

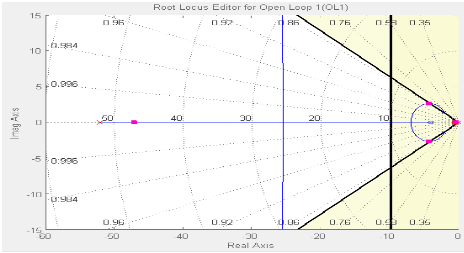

Fig. 6. Root locus of the PID controller.

There are 4 poles in Fig. 6 that represent the stability of the system. The most of the poles are located very close to the imaginary axis so that the system has a decreasing value of rise time. There are two poles that are not located on real axis and imaginary axis. They indicate that there is an overshoot for the system. If the poles pass through the imaginary axis then the system becomes unstable. Overall based on the graph, the system is still in stable condition [5], [6].
The unit step is given for PID controller design. Fig. 7 shows cares ponding response of the system.

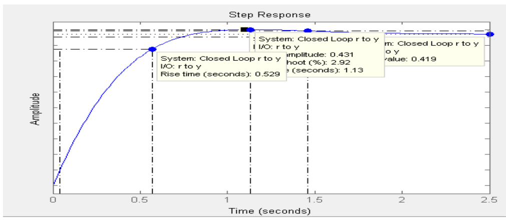

Fig. 7. Unit step response of the PID controller.

Based on Fig. 7 we can get transient parameters for the system such as rise time, percent overshoot, settling time, and steady state error. For the system, we can get 0, 52 second for the rise time (Tr), 2,92% for overshoot (Mp), 1,46 second for the settling time (Ts), and 4.1% steady state error (Ess). After the design and analyze of PID controller response, the Simulink model is shown in Fig 8.

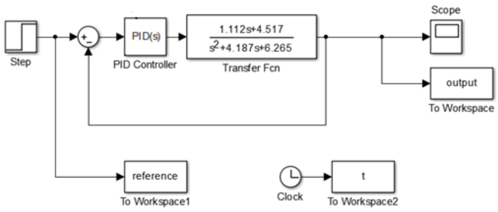

Fig. 8. Simulink model for simulating the PID controller.

Based on the Simulink model of the PID controller, the result of simulation is shown in Fig. 9.

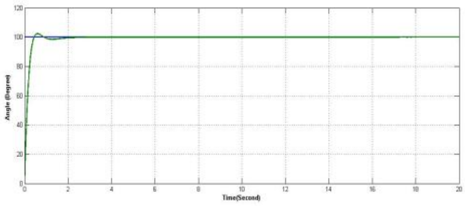

Fig. 9. Simulation result of the PID controller.

Fig. 10 shows Simulink model implementation of the PID controller to the Arduino Mega 2560 via potentiometer by using Simulink Support Package for Arduino Hardware.

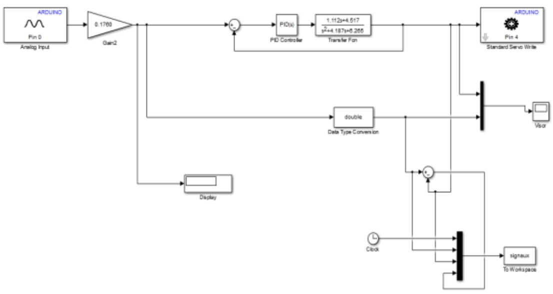

Fig.10. Simulink model for implementing the PID controller.

Based on the Simulink model for implementing PID controller, Fig. 11 shows the obtained result of implementation.

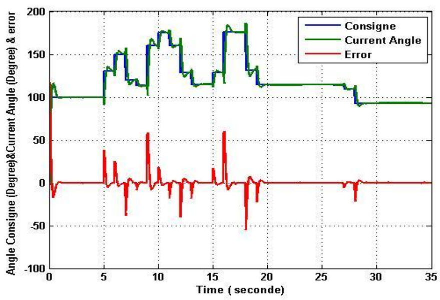

Fig. 11. Implementation result of the PID controller.

III. ACO-PID CONTROLLER

In the control system, the PID control system is more popular for setting the parameters. The conventional fixed gain PID regulator is a well-known technique for most industrial control processes. The design of this controller requires the three main parameters, namely the proportional gain (Kp), the integral time constant (Ki) and the derived time constant (Kd). This controller gain is adjusted by a test and error method based on the history and behavior of the installation. This process will require more time and will only be suitable for particular operating conditions, which means that when the fixed gain is present the error value always remains the same at this condition. In this project, the ACO algorithm is used to optimize the gains, and the values are transferred to the PID controller of the installation representing the AVR of the power generation system, as shown in the figure.12. The proportional gain causes the controller to respond to the error value while the integral gain helps eliminate steady-state errors and derivative gain to prevent overruns. The plant is replaced by AVR models developed with Simulink in MATLAB.

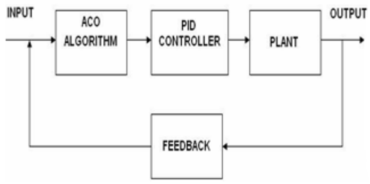

Figure 12: ACO-PID Controller .

Table I: ACO Parameter Specifications

ACO Properties	*Properties*
Type of ACO	Ant Colony System
Population of ants	20
Maximum length of path	8
Pheromone coefficient , β	5
Heuristic coefficient ,α	5
Evaporation rate ,ρ	0.5
Convergence condition	Max-min of 20 pop≤0.0001
Maximum Iteration	40

After designing and building the ACO-P & ID, Figure 13. shows the Simulink model to simulate ACO-PID.

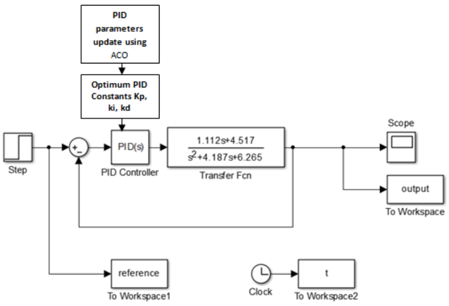

Fig.13. Simulink model for simulating ACO-PID

Based on the Simulink model for ACO-PID simulation, the obtained result is shown in Fig 14

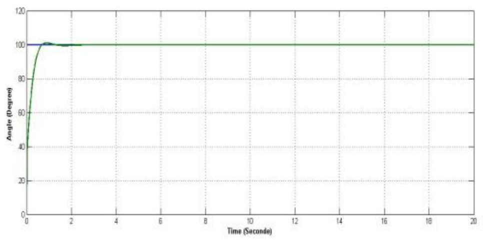

Fig. 14. Simulation result of the ACO-PID.

Based on Fig. 14, we can get transient parameters of the system. They are a rise time, percent overshoot, settling time, and steady state error. For the system the obtained values are 0, 09 second for the rise time (Tr), 1% for overshoot (Mp), 0,89 second for the settling time (Ts), and 0 % steady state

The time response parameters, including rise time (Tr), percentage overshoot (Mp), settling time (Ts), and steady state error (Ess) of the PID controller and the ACO controller PID for the higher order system transfer function of DC servo motor are presented in Table II.

TABLE II. Time response parameter.

Controller Type	**Tr (Sec)**	**Mp (%)**	**Ts (Sec)**	**Ess (%)**
PID controller	0.529	2.92	1.46	4.1
ACO-PID	0.09	0.0107	0.898	0

VI. CONCLUSION

The design of the DC servo motor system using both the PID controller and the ACO-PID were presented. Design of the PID controller has been implemented successfully, but it still has one overtake. Thus, the best control design has been proposed to control a smooth precision movement of the manipulator arm. Better control of performance, robustness and overall stability can be expected using ACO-PID.With the tuning method using ant colony optimization, the controllers have better stability, a little overshoot and a quick response. Based on the results, the time response parameters can be observed. The PID controller provides a very large steady state error due to oscillatory behavior in transient period. It has severe oscillations with a very high overshoot of 2.92% the damage in the system performance. The proposed ACO-PID can effectively eliminate these dangerous oscillations and provides smooth operation during the transient period. Therefore, it is concluded that the PID controller could not be used for the accuracy of the movement control of the manipulator arm model because of the oscillator.

The results of the simulation show that the ACO has a process of self-adaptation and positive feedback to get out of the trap of a local minimum and get the best solutions. The overall performance of the system is improved and the requirements are met.

References:

[1] A. Visioli, “Tuning of PID controllers with fuzzy logic,” IEE Proceedings: Control Theory and Applications, vol. 148, no. 1, pp. 1–8, 2001.**doi**:10.1049/ip-cta:20010232

[2] Gaurav, Amrit Kaur, “Comparison between Conventional PID and Fuzzy Logic Controller for Liquid Flow Control: Performance Evaluation of Fuzzy Logic and PID Controller by Using MATLAB/Simulink,” International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-1, Issue-1, June 2012. **doi**=10.1.1.676.4147

[3] A. Oonsivilai and P. Pao-La-Or, “Application of adaptive tabu search for optimum PID controller tuning AVR system,” WSEAS Transactions on Power Systems, vol. 3, no. 6, pp. 495–506, 2008

[4] C. Urrea and J Kern, "A New Model for Analog Servo Motors", Canadian Journal on Automation, Control and Intelligent Systems, Vol. 2, No. 2, March 2011.

[5] S.R.Vaishnav, Z.J.Khan, "Design and Performance of PID and Fuzzy Logic Controller with Smaller Rule Set for Higher Order System," Proceedings of the World Congress on Engineering and Computer Science, WCECS 2007, October 24-26, 2007, San Francisco, USA.

[6] C. Cao, X. Guo, and Y. Liu, "Research on ant colony neural network PID controller and application," in Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '07), pp. 253–258, 2007. **doi:** 10.1109/SNPD.2007.308

[8] P. Wang and D. P. Kwok, "Optimal design of PID process controllers based on genetic algorithms," Control Engineering Practice, vol. 2, no. 4, pp. 641–648, 1994.https://doi.org/10.1016/0967-0661(94)90008-6

[9] M. Dorigo and G. Di Caro, "The ant colony optimization meta-heuristic," in New Ideas in Optimization, D. Corne, M. Dorigo, and F. Glover, Eds., pp. 11–32, McGraw Hill, London, UK, 1999. **doi :** 10.1109/CEC.1999.782657

[10] Ayman A. Aly, "PID Parameters Optimization Using Genetic Algorithm Technique for Electrohydraulic Servo Control System", Intelligent Control and Automation, 2011, 2, 69-7.**doi**: 10.4236/ica.2011.22008

[11] Arturo Y. Jaen-Cuellar,'' PID-Controller Tuning Optimization with Genetic Algorithms in Servo Systems'', *International Journal of Advanced Robotic Systems (IJARS),* September11, 2013. https://doi.org/10.5772/56697

[12] **.**D. C. Meena ; Ambrish Devanshu'' Genetic algorithm tuned PID controller for process control'', International Conference on Inventive Systems and Control (ICISC),19-20 Jan. 2017, **doi:** 10.1109/ICISC.2017.8068639

[14] P. Cominos and N. Munro, "PID Controllers: Recent Tuning Methods and Design to Specification," IEE Proceedings Control Theory & Applications, Vol. 149, No. I, January 2002.**doi:** 10.1049/ip-cta: 20020103

[15] J. G. Ziegler and N. B. Nichols, "Optimum settlings for automatic controllers," Transactions of the ASME, vol. 64, pp. 759–768, 1942. **doi:**10.1115/1.2899060

[16] Kim T. H., Maruta I., Sugie T., Robust PID controller tuning based on the constrained particle swarm optimization, Automatica, Vol. 44, Issue 4, Apr. 2008, p. 1104 – 1110, https://doi.org/10.1016/j.automatica.2007.08.017

[17] Munadi, M. Amirullah Akbar, '' Simulation of Fuzzy Logic Control for DC Servo Motor using Arduino based on Matlab/Simulink'', International Conference on Intelligent Autonomous Agents, Networks and Systems Bandung, Indonesia, August 19-21,2014,
doi**:** 10.1109/INAGENTSYS.2014.7005723

[18] S. F. Abd-Alkarim "Application of Fuzzy Logic in Servo Motor", Al- Khwarizmi Engineering Journal 2007, Vol.3, No.2, pp. 8 -16. , **ISSN**: 18181171 23120789

[19] Jihong Li, Pingkang Li Stability Region Analysis of PID Controllers for Time-delay Systems", proceeding of the 6thWorld Congress on Intelligent Control and automation, 21-23 (June 2006), pp. 2219-2223, **doi:** 10.1109/WCICA.2006.1712753

[20] Koivo H N, Tanttu J T. Tuning of PID controllers: survey of SISO and MIMO techniques[C]//Proceedings of the IFAC Intelligent Tuning and Adaptive Control Symposium. 1991: 75-80. https://doi.org/10.1016/S1474-6670(17)51299-9.