Comparison for Control Robotic Arm Using MIGO and PSO Algorithms

Mahmood M Olwane[1], Ahmed F.Mahmoud Zrigan[2], Salem Alarabi Shufat[3]

Department of Electrical and Electronic Engineering-Higher Institute of Engineering Techniques -Tripoli[1,3]

Department of Electrical and Electronic Technology-Higher Institute of Science and Technology-Tripoli[2]

amwajal@hotmail.com[1] ,AhmedZrigan@yahoo.com[2] , s.shffat@hiett.edu.ly[3]

Abstract
It is currently vitally need to conduct research and build considerable efforts for Robot or the robotic arm of the lower volume(small size, high quality, high efficiency). MATLAB software was used to design an articulated robotic arm test system, construct mathematical models for the system, perform numerical simulations, and perform PID optimization on the system. Simulation results demonstrate that, in accordance with synchronized control and PID optimization, this system has the advantages of strong driving capability for the hydraulic drive system as well as high positioning accuracy for the motor drive system. It also demonstrated high tracking accuracy and high speed of response for the input signal of a large load and has realized a combination servo drive of a smaller amount heavy articulated robotic arm of fast high-precision.

Keywords: SISO toolbox, PID controller, MIGO Step Response Methods, Particle Swarm Optimization (PSO) algorithm, MATLAB\SIMLUNK.

الملخص
في الوقت الحالي، هناك حاجة حيوية لإجراء البحوث وبناء الجهود الكبيرة للروبوت والذراع الروبوتية ذات الحجم الصغير (حجم صغير، جودة عالية ، كفاءة عالية). تم استخدام برنامج MATLAB لتصميم نظام اختبار مفصلي للذراع الآلية ، وإنشاء نماذج رياضية

للنظام ، وإجراء عمليات محاكاة عددية ، وإجراء تحسين PID علي النظام . توضح نتائج المحاكاة أنه وفقًا للتحكم المتزامن وتحسين PID , يتمتع هذا النظام بمزايا قدرة القيادة القوية لنظام الدفع الهيدروليكي بالإضافة إلى دقة تحديد المواقع العالية للنظام . أظهر أيضًا دقة تتبع وسرعة استجابة عالية لإشارة الدخل.

الكلمات المفتاحية: صندوق أدوات SISO، وحدة تحكم PID، طرق استجابة خطوات MIGO ، خوارزمية تحسين سرب الجسيمات (PSO)، MATLAB \ SIMLUNK.

I. INTRODUCTION

Things considered, robotics refers to the research, design, and application of robot systems in manufacturing. Robots are typically employed to carry out unpleasant, risky, extremely repetitive, and harmful activities [1]. DC motors are the most often utilized actuator in mechatronic motion control applications. Although there are numerous resources that offer various design and selection strategies for controlling motions in the desired manner, the majority of control systems in use are based on conventional PID controllers [2]. Robots are increasingly being used because they are seen as essential components of automation [1]. Robotic autonomy and intelligence are essentially the result of technological developments and research in fields like modeling, design, control, and artificial intelligence (AI) [2]. The scientific community is becoming increasingly interested in modeling and simulation in various fields of study in order to get a deeper knowledge of practical applications. The more an anthropomorphic system is studied and modeled, the better. Knowledge of basic human biomechanics, as well as lead to the creation of biological agent control laws.

The feedback loop serves as the foundation for creating the control method for a robotic manipulator and is crucial in dampening uncertainty. The plant's control system eliminates a variety of interruptions and uncertainties. In order to accomplish the desired behaviour, the control problem must be solved by defining input signals like velocity or actuator input voltage. The controller must handle all complexity, dynamic coupling, and nonlinearities. The aforementioned problems cannot be solved by simple linear control law-based solutions [3].

The development of the Linear Quadratic Gaussian control (LQG) in the 1960s marked the maturation of the optimal control, which was based on Wiener's optimal filtering work [10]. A linear model of the process at a certain operating point must be obtained in order to synthesize the LQG control. The Ziegler-Nichols approach and MIGO (M - constrained integral gain optimization) have comparable processes [3].

The constant for this method is often calculated using the equations provided by ASTRÖM and HÄGGLUND [1] [2]. In this study, the SISOTOOL in MATLAB/Simulink is used to construct the AMIGO method [3].

The simulation and control of Robot arm is the main purpose of this paper. For this purpose, a brief introduction about Robotic Arm system description is including equations modeling, MIGO Approximation algorithm, and PSO algorithm. Then view the simulation and results. In addition, final the Conclusions.

II. ROBOTIC ARM SYSTEM DESCRIPTION

Motion control can be reduced to DC motor motion control because DC motors are the most common actuator used in mechatronic motion applications. A PMDC motor is an example of an electromechanical system that combines mechanical and electrical components. A simplified equivalent representation of the two components of an armature-controlled PMDC motor is illustrated in Figure 1 together with a mobile robot and a single jointed robot arm in Figure 2 [4].

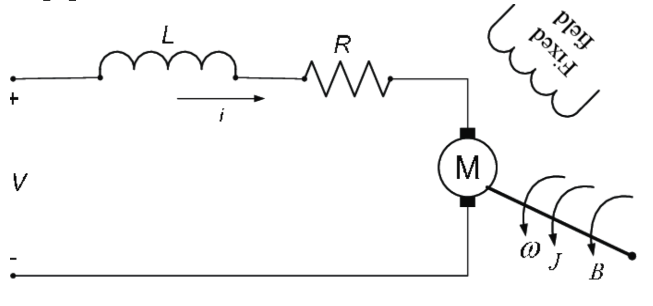

Fig 1: Electromechanical PMDC motor component equivalent representation.

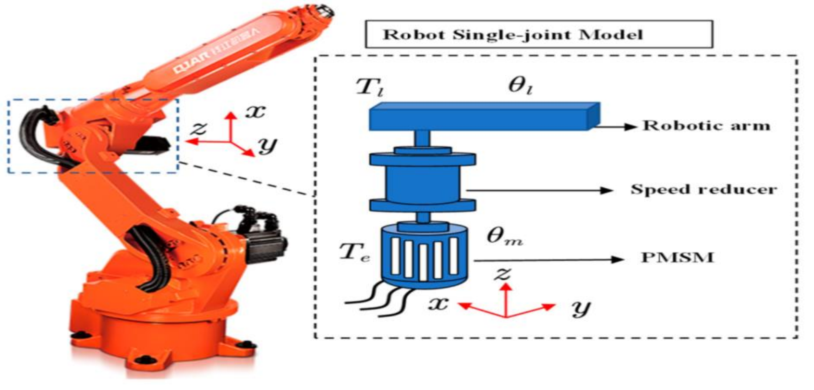

Fig 2: Single joint robot arm.

A. equations modeling

The motor torque is given by Eq (1). The generated EMF voltage, e_a, is given by Eq (2).

$$T_m = K_t i_a \qquad (1)$$

$$e_a = K_b \omega_m = K_b \frac{d\theta_m}{dt} \qquad (2)$$

Eq(3) is produced by placing Kirchoff's law around the electrical loop. Laplace transform and rearrangement yield Eq. (4):

$$l_a \frac{di}{dt} + R_a i = V - K_e \frac{d\theta_m}{dt} \qquad (3)$$

$$I_a(s) = \frac{V(s) - K_e S\theta_m(s)}{R_a + l_a S} \qquad (4)$$

When the torques are added together during the energy balance, the result is Eq. (5), Eq. (6) is produced by using the Laplace transform and rearranging [4].

$$K_t * i - T_{Load} + J_m \left(\frac{d^2\theta}{dt^2}\right) - b_m \left(\frac{d\theta}{dt}\right) = 0 \qquad (5)$$

$$K_t I(s) = (J_m S + b_m) S\theta(s) \tag{6}$$

The transfer function is provided by Eq.(7) in terms of the input voltage, V(s), and the output motor shaft angleθ_m, after inserting Eq.(4) in Eq.(6) and rearrangement.

$$G_m = \frac{\theta_m(s)}{V(s)} = \frac{K_t}{S[(R_a + L_m S)(J_a s + b_m) K_b K_t]} \tag{7}$$

At the motor's armature, the estimated total inertia, J_{eq}, and damping, b_{eq}, are calculated as follows:

$$b_{eq} = b_m + \frac{b_{load}}{n^2}, \qquad J_{eq} = J_m + \frac{J_{load}}{n^2}$$

The entire equivalent transfer function, linking input voltage V_{in} and Arm-load output angular position θ_{load}, is obtained by Eq. (7) after substituting J_{eq} and b_{eq} (8). This transfer function includes the gear system's transfer function, the gear ratio (n).

$$G = \frac{\theta_{load}(s)}{V_{in}(s)}$$

$$G = \frac{K_t * n}{L_a J_{eq} s^3 + (R_a J_{eq} + b_{eq} L_a) s^2 + (R_a b_{eq} + K_t K_b) S} \tag{8}$$

The arm mass, M, is 8 kg, the arm length, L, is 0.4 m, and the viscous damping constant, b, is 0.09 N.sec/m. These numbers are the nominal parameters for the robot arm system that needs to be designed. The nominal values listed below were used for the eclectic motor's various parameters: V_{in}=12 volts, J_m=0.02 kilograms per square meter, b_m=0.03, K_t=0.023 N-m/A, K_b=0.023 V-s/rad, R_a=1 ohm, L_a=0.23 henry, and TLoad, gear ratio, with n=1 for simplicity.Eq. (9) is produced when parameter values are substituted into the transfer function [4].

$$G = \frac{0.023}{0.02913 s^3 + 0.1543 s^2 + 0.1205 s} \tag{9}$$

Regarding state space specification A, B, C and D matrix could be obtain as below Eq. (10):

$$X^{\cdot} = \begin{bmatrix} -5.2969 & -4.1366 & 0 \\ 1.0000 & 0 & 0 \\ -0 & 1.0000 & 0 \end{bmatrix} X + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} U \qquad (\mathbf{10})$$

$$Y = [0 \quad 0 \quad 0.7896] X + [0]$$

B. MIGO Approximation algorithm

M-constrained Integral Gain Optimization with approximations. Using straightforward process models approximate the MIGO design. Determined by step or frequency responses, there are three requirements, Adjusting controller settings to a sizable test batch, implementing basic tuning guidelines like Ziegler-Nichols [5-6].

C. PSO algorithm

Particle update rule

$$p = p + v$$

with

$$v = v + c_1 * rand * (p_{Best} - p) + c_2 * rand * (g_{Best} - p)$$

Where

p: particle's position

v: path direction

c_1: Weight of local information

c_2: Weight of global information

p_{Best}: Best position of the particle

g_{Best}: Best position of the swarmrand: random variable

Number of particles usually between 10 and 50, C1 is the importance of personal best value; C2 is the importance of neighborhood best value. Usually C1+ C2 = 4 (empirically chosen value). If velocity is too low → algorithm too slow. If velocity is too high → algorithm too unstable, create a 'population' of agents (particles) uniformly distributed over X. Evaluate each particle's position according to the objective function. If a particle's current position is better than its previous best position, update it. Determine

International Science and Technology Journal
المجلة الدولية للعلوم والتقنية

Volume31 العدد
November 2022 نوفمبر

المجلة الدولية للعلوم والتقنية
International Science and Technology Journal
ISTJ

the best particle (according to the particle's previous best positions) [6].

Update particles' velocities:

$$v_i^{t+1} = \underbrace{v_i^t}_{\text{inertia}} + \underbrace{c_1 U_1^t \,(pb_i^t - p_i^t)}_{\text{personal influence}} + \underbrace{c_2 U_2^t \,(gb^t - p_i^t)}_{\text{social influence}}$$

Move particles to their new positions:

$$p_i^{t+1} = p_1^t + v_i^{t+1}$$

Until stopping criteria are satisfied

Particle's velocity:

$$v_i^{t+1} = \underbrace{v_i^t}_{\text{Diversification}} + \underbrace{c_1 U_1^t \,(pb_i^t - p_i^t) + c_2 U_2^t \,(gb^t - p_i^t)}_{\text{Intensification}}$$

III. Simulation and Results

It is preferable to have knowledge of how different coefficients Kp, Ki, and Kd affect the system before utilizing the PID command. Table 1 provides an overview of how each controller parameter, Kp, Ki, and Kd, affects a closed-loop system.

Table1. Effects of PID controller parameters

CL Response	RiseTime	Overshoot	Settling Time	S-S Error
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small Change	Decrease	Decrease	No change

Comparison of Step Response Characteristics between Approximation MIGO and PSO Algorithm showing in Figure 3.

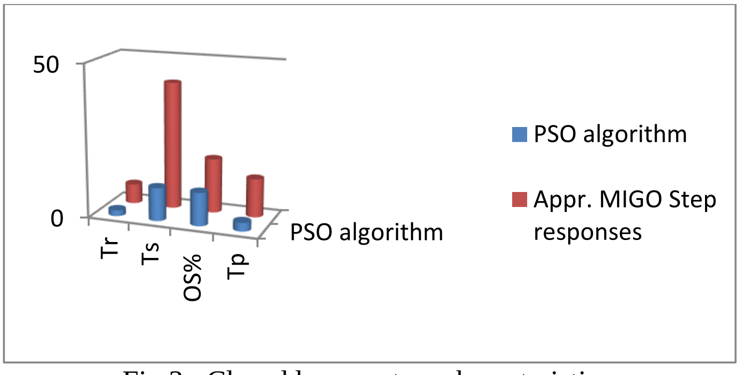

Fig 3: Closed loop system characteristics.

To show the effectiveness of the PSO algorithm, a comparison is made with the designed PID controller with PSO algorithm and MIGO algorithm. Figure 4 shows step response of the system after their applying.

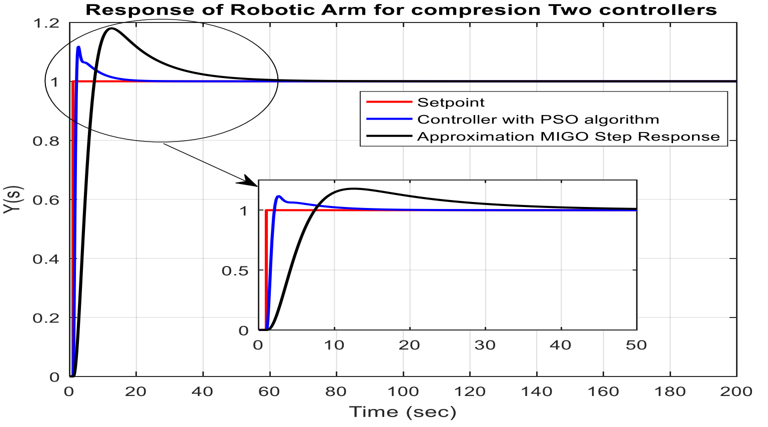

Fig 4: The step response of system using MIGO algorithm and PSO algorithm.

From the figure shown above, it is clear using PSO algorithm is better than MIGO algorithm to achieve stability and fast response. Finally, this result is only preliminary research. To further investigate the effectiveness of the PSO algorithm, work may be done such as Comparison of the PSO-PID with other tuning method.

IV .Conclusions

MATLAB software was used in this paper to simulation articulated robotic arm system, construct mathematical models for the system, and perform PID optimization on the system using MIGO algorithm and PSO algorithm ,simulation showed that, It was shown that PSO is a suitable . Therefore, the PSO algorithm has more robust stability and efficiency, and solves tuning problems of PID controller parameters. Where the results were compared in both cases, PSO was able to converge well as high positioning accuracy for the motor drive system.

REFERENCES

[1] Venkata Satya Durga Manohar Sahu; Padarbinda Samal; Chinmoy Kumar Panigrahi , "Implementation of Particle Swarm Optimization for Solving Optimal Control Problem of a Flexible Robotic Manipulator" i 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), 26-28 August 2022,Ravet, India.

[2] R. Agrawal, "Modeling a Controller for an Articulated Robotic Arm", Intelligent Control and Automation, 2012, 3, pp. 207-210.

[3] "Particle swarm optimization" , 8 February 2023 . In Wikipedia . last seen 2/25/2023. https://en.wikipedia.org/wiki/Particle_swarm_optimization.

[4] "Particleswarm" , 2014 . In Mathworks . last seen 2/25/2023 . https://en.wikipedia.org/wiki/Particle_swarm_optimization.

[5] F. A. Salem, "Modeling, Simulation and Control Issues for a Robot ARM; Education and Research", I.J. Intelligent Systems and Applications, 2014, 04, pp.26-39.

[6] Andersson & Lindberg , 2004 . "Autotuning of a PID-controller" . last seen 2/24/2023 . https://www.lunduniversity.lu.se/lup/publication/8848024.