Journal of Futuristic Sciences and Applications
Vol. 7(1), Jan-Jun 2024, pp. 8-13
doi: 10.51976/jfsa.712402
www.gla.ac.in/journals/jfsa

Enhancing Robot Dynamics Control for Accuracy and Effectiveness using Genetic Algorithms

Rasiyawan Yadav and Preet Lata***

ABSTRACT

In this paper evaluated the effectiveness of a genetic algorithm (GA) in reducing error in robot dynamics control. Experiments demonstrate that GAs are useful in determining the optimal robot designs, yielding results with lower error rates and higher precision. Notably, the method just requires position feedback and the final equations of the dynamic model's links, eliminating the need for speed and acceleration data, which typically lead to significant identification errors. Using the advances in computer technology, this deficit might eventually disappear, albeit it could take longer. Future studies should include cost considerations in the objective function to solve the deterministic and random faults in the model.

Keywords: *GA, Optimization, Intelligent Control, Metaheuristic, Modelling, Nonlinear.*

1.0 Introduction

Robot industrial applications have grown in popularity since their invention. Robots must be exact and consistent in a variety of applicationsThe capacity of a robot to return to the same position again and over again is called repeatability. Accuracy is the ability of a robot to move correctly to a specified posture in three dimensions. Robot connections and joint angle are responsible for the accuracy of robots; nevertheless, the statistical importance of robot parameter tolerances and the correlations among these parameters have not yet been examined. It is uncommon for a simulation technique to be developed in order to study robot parameter tolerance, despite the fact that several scholars have attempted to examine the effects of various robot parameters on performance [1-6]. To attain the necessary accuracy in robot performance, the optimum tolerance design approach takes into account the link size, joint angles, and torque generated by each robot part. Robotic arm trajectory optimization is a common design difficulty. Because of the complexities of this work in the past, many of the proposed solutions were mediocre at best. As a result, many authors have previously used evolutionary algorithms. EA was first used for collision-free robotic arm path planning in 1997. Following this, the concept and application of Genetic Algorithms and Simulated Annealing for calculating the optimal trajectory of a multiple robot setup entered research. The Evolution Approach for optimal trajectory control states that the optimal trajectory based on cubic polynomials is computed in the first stage under certain physical constraints. After doing this research in the aim of an overview of the usage of evolutionary algorithms in controller design and robotics, Pires offers a Genetic Algorithm for constructing manipulator trajectories that include obstacle avoidance.

**Manager, National Thermal Power Corporation Limited, New Delhi, India*
***Corresponding author; Assistant Professor, Department of Electrical Engineering, NIT Kurukeshtra, Kurukeshtra, Haryana, India (E-mail: preetklata@gmail.com)*

The widely utilized GA-based method for five-degrees-of-freedom robot parameter identification is thoroughly studied. In this scenario, a model reflects the model of the robot's course at each instant in time with tracking defects. As a result, if the parameters are chosen in such a way that the model with these parameters accurately reproduces the robot's position feedback along the same course, the robot may be reliably identified. As a result, robot parameters can be totally determined by location response data. For identifying model parameters that provide the same type of position response in time as the robot for the trajectory, genetic algorithms are a viable search strategy. The methodology for simulating the real-world performance of a robot with noise effects was investigated in this study for robot parameter tolerance combinations, and GAs were used to demonstrate the selection of optimal tolerance criteria. Finally, the simulation's findings. To show the suggested methodology, a two-link stiff robotic manipulator is validated using simulation data [7-10]. The suggested method for optimal tolerance selection is inexpensive, requiring no capital investment in simulation equipment and incurring only minor computing costs. Prior to costly manufacture, the suggested endeavor would assist robotic system designers in making parameter tolerance criterion judgments [11-12].

2.0 Manipulator Modelling

Robot manipulators may be mathematically characterized using a variety of methods. The problem of kinematics is to characterize the motion of the manipulator without taking torques and forces into account. These formulae determine the position and orientation of the end effector based on the values of the joint variables and the position and orientation of the end effector based on the values of the joint variables. Dynamic modeling requires the development of equations that clearly explain the relationship between force and motion.These are the formulae that are needed to simulate robot motions and create control systems.

Although the framework is intended for manipulators with stiff links and just revolute joints, it may be adjusted to support different types of manipulators. Because of the recursive technique, the described composition for three degrees of freedom is easily extensible to any number of degrees of freedom. The equations can be simplified and a dynamic model as been designed on MATLAB Simulink here tau represents the torque and theta is referring as an input rest all are parameters required for dynamic modelling of 2 link manipulator.

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_{11} \\ \ddot{\theta}_{22} \end{bmatrix} + \begin{bmatrix} P_{11} \\ P_{21} \end{bmatrix} + \begin{bmatrix} f_{r1} \\ f_{r2} \end{bmatrix} + \begin{bmatrix} fn_{1p} \\ fn_{2p} \end{bmatrix} = \begin{bmatrix} \tau f_{1p} \\ \tau f_{2p} \end{bmatrix} \quad \ldots 1$$

Subsystems may be evaluated independently since large dynamic systems are the result of complex manipulators with several degrees of freedom. All of the inertia matrix's constituents, along with the gravity and Coriolis elements, are evaluated separately at the framework's conclusion. With the MATLAB conversion code at the end, you may use this simple approach of translating Maple code to MATLAB code to any expression. It should be noted that the MATLAB conversion algorithm does not support joint variable derivatives. To alter these values as needed, use MATLAB. It is not necessary to deny symbolic values if a less complex model is the desired outcome; nevertheless, bear in mind that the bigger the dynamic model, the more symbolic values and zeros denied. The dynamic model has produced the following final equations-

$$\ddot{\theta}_{11} = \frac{\tau f_{1p} - f_{n1p} - f_{r1} - P_{11} - S_{12} * \ddot{\theta}_{22}}{S_{11}} \quad \ldots 2$$

$$\ddot{\theta}_{22} = \frac{(\tau f_{2p} - f_{n2p} - f_{r2} - P_{21} - S_{12} * \ddot{\theta}_{11})}{S_{22}} \quad \ldots 3$$

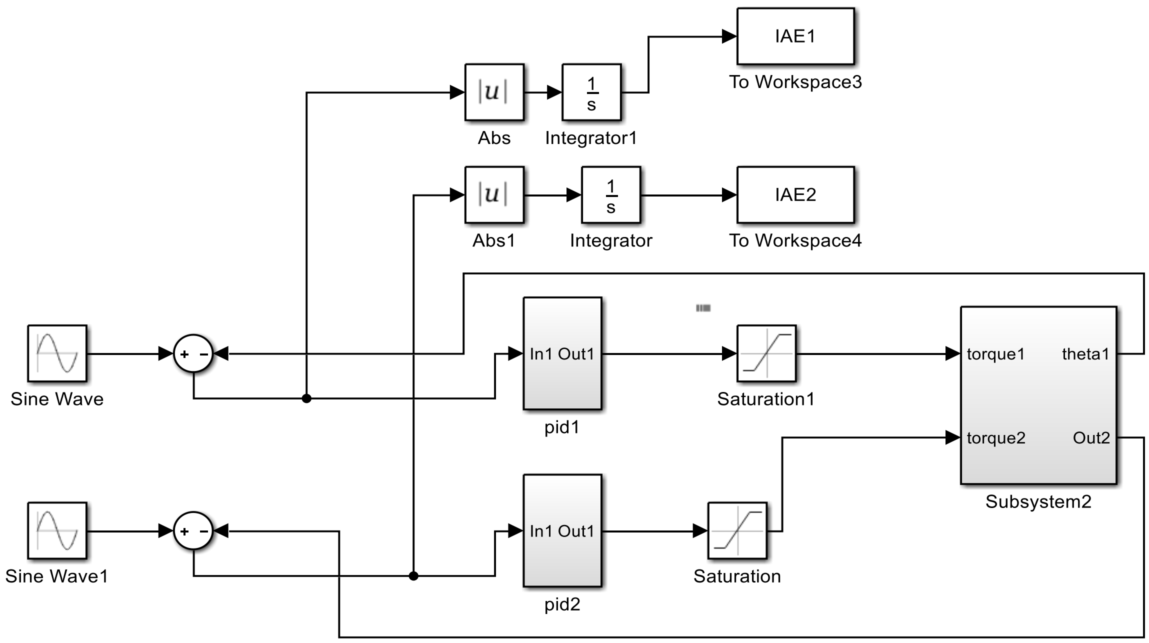
Doi: 10.51976/jfsa.712402

3.0 Optimization through Genetic Algorithm

Darwin's theory of evolution, which models the survival of genetically more fit creatures, served as the inspiration for GA. Population-based algorithms like the GA algorithm are used. A chromosome is represented by each solution, and a gene is represented by each parameter. GA measures each person's fitness within the population using a fitness function. The optimal answers are selected at random via a selection mechanism in order to improve flawed solutions. Due to the proportionality of probability, this operator has a higher likelihood of choosing the optimal solutions. The probability of avoiding local optima is increased when poor solutions are an option. This implies that one can use other solutions to derive good solutions from a local solution. The exploitation of the "region" between the two parent solutions results from crossing individuals.

Also helpful to this approach is mutation. This operator preserves population variety and increases GA's curiosity by randomly modifying the genes on the chromosomes. The mutation operator, like nature, may result in a significantly different outcome. Improved solution that will lead to the global optimum of alternative options A population is first created at random using the GA technique. To boost the diversity of this population, a Gaussian random distribution can be employed to build it. This population consists of numerous solutions that represent various people's chromosomes. On each chromosome, there are genes-simulating variables. The primary purpose of the initialization phase is to distribute solutions as uniformly as possible across the search space in order to maximize population diversity and the likelihood of finding interesting sites. The most fit people have a better chance of acquiring food and mating in the natural environment. This boosts their genes' contribution to the next generation of the same species. The GA algorithm assigns probability to persons and selects them for the next generation based on their fitness scores using a roulette wheel.

Figure 1: Optimized Controller Diagram Design in SIMULINK

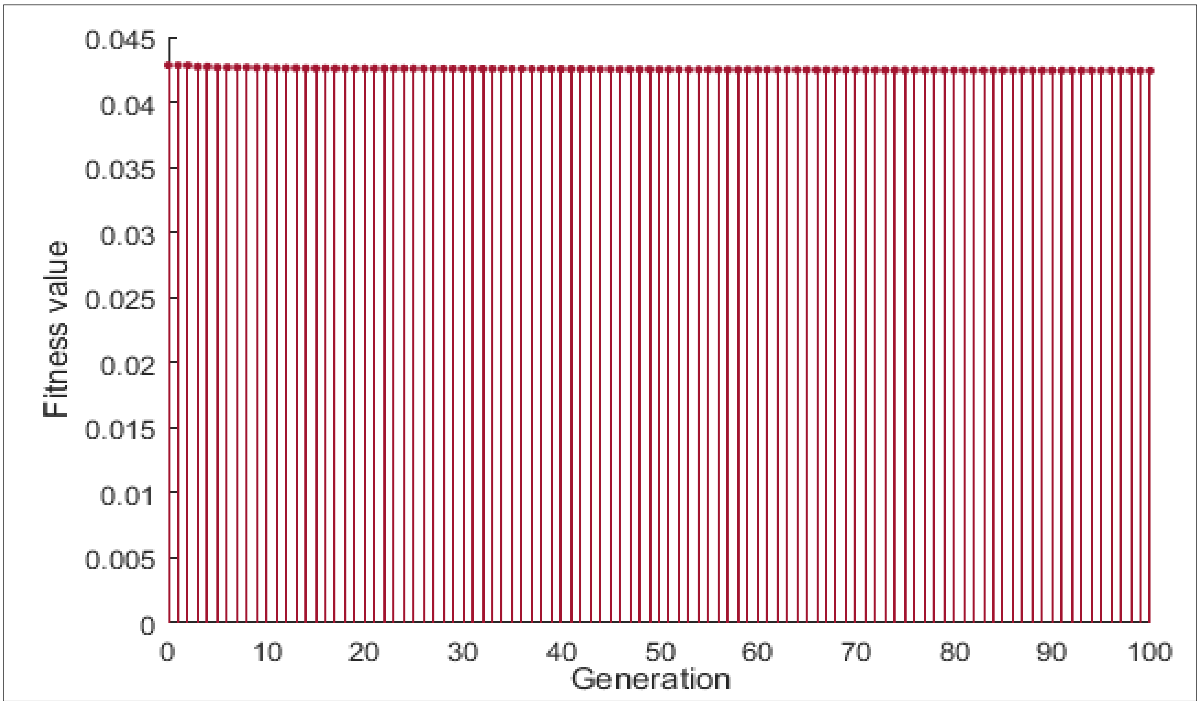


The figure above shows the optimized controller on which GA has been applied, the dynamic model described above is being treated here as a plant on which a controller is attached here Integral absolute error has been chosen as an objective function. The GA algorithm starts with a population of people chosen at random. This method improves the population by employing the three aforementioned operators until the end condition is met. The best answer from the most recent population is returned as the best approximation of the global optimal solution for a particular problem. The rates of selection, crossover, and mutation can be changed or fixed throughout optimization.

4.0 Results

Using the optimal control parameters for each connection angle yields the desired results after 120 algorithm iterations with a population size of 100. Despite the fact that both connections' beginning circumstances differ from the starting points of the monitored trajectories, as these figures show, their responses have tracked the planned trajectories without overshooting for the given parameters, and both errors have rapidly converged to zero. It was demonstrated that GAs could efficiently generate results and identify robot settings. Performance is also seen to be impacted by variations in the initial state, whereas other dynamic elements have less influence. The graphs are shown in Figure.

Figure 2: Graph Representing Least Value of Error after Optimization



The above figure shows the minimum value of error that is 0.043 after running the GA through 100 generations the generation wise improvement in error is so smooth that can be easily visible in the graph that has been represented in the form of stem. here both the axis represents the fitness value in points and no of generations that has been runned in order to get optimized result for the controller. This algorithm is dependable and capable of predicting the global ideal for a specific problem due to its strategy of retaining the best solutions in each generation and applying them to improve subsequent solutions. As a result, from generation to generation, the population improves.

5.0 Conclusion

Through its ability to emulate biological evolution and the survival of the fittest, Genetic Algorithms (GAs) provide a viable solution to complex optimization problems. In this study, we introduced the Genetic Algorithm (GA) technique and evaluated its efficacy in determining the optimal control structure for a certain time horizon within the context of a control issue. The significance of identifying various control factors is underlined, especially for nonlinear systems where conventional methods are unclear and, hence, ineffectual. This paper makes a significant addition by proposing the use of genetic algorithms as a metaheuristic optimization technique to solve the challenge of identifying the ideal nonlinear controller settings.

Using this novel approach on a dynamic model of a stiff two-link robot manipulator, our findings confirm that the recommended optimal control strategy is effective for both regulation and trajectory tracking problems. The promising results obtained by combining intelligent control techniques with GA-based optimization demonstrate the potential of this strategy in addressing control challenges in complex systems. In addition to demonstrating the efficiency of genetic algorithms in identifying nonlinear control parameters, this study encourages a broader application of metaheuristic optimization techniques to address complicated control problems across a range of domains. As a result, our work advances the theoretical understanding of control optimization and provides helpful guidance for developing trustworthy control strategies for real-world applications.

References

[1] Yang, H., Liu, H., Zou, J., Yin, Z., Liu, L., Yang, G., Ouyang, X. and Wang, Z. eds., 2023. Intelligent Robotics and Applications: 16th International Conference, ICIRA 2023, Hangzhou, China, July 5–7, 2023, Proceedings, Part II (Vol. 14268). Springer Nature.
[2] Ge, S.S., Lee, T.H. and Zhu, G., 1996. Genetic algorithm tuning of Lyapunov-based controllers: An application to a single-link flexible robot system. IEEE Transactions on Industrial Electronics, 43(5), pp.567-574.
[3] Bakdi, A., Hentout, A., Boutami, H., Maoudj, A., Hachour, O. and Bouzouia, B., 2017. Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. Robotics and Autonomous Systems, 89, pp.95-109.
[4] Deif, S., Kamal, H.A. and Tawfik, M., 2012. Enhancing genetic algorithms using a dy-namic mutation value approach: An application to the control of flexible Robot Systems. International Journal of Artificial Intelligent Systems and Machine Learning, 4(1), pp.9-16.
[5] Mucientes, M., Moreno, D.L., Bugarín, A. and Barro, S., 2007. Design of a fuzzy controller in mobile robotics using genetic algorithms. Applied Soft Computing, 7(2), pp.540-546.
[6] Wang, B., Fang, J., Qi, S., Wang, L., Liu, X. and Ren, H., 2023. Step-by-step identification of industrial robot dynamics model parameters and force-free control for robot teaching. Journal of Mechanical Science and Technology, 37(7), pp.3747-3762.

[7] Jaen-Cuellar, A.Y., de J. Romero-Troncoso, R., Morales-Velazquez, L. and Osornio-Rios, R.A., 2013. PID-controller tuning optimization with genetic algorithms in servo systems. International Journal of Advanced Robotic Systems, 10(9), p.324.

[8] Qu, H., Xing, K. and Alexander, T., 2013. An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. Neurocomputing, 120, pp.509-517.

[9] Costa, G., Pinho, J., Botto, M.A. and Lima, P.U., 2023. Online learning of MPC for autonomous racing. Robotics and Autonomous Systems, 167, p.104469.

[10] Shi, P. and Cui, Y., 2010, May. Dynamic path planning for mobile robot based on genetic algorithm in unknown environment. In 2010 Chinese control and decision conference (pp. 4325-4329). IEEE.

[11] Juang, C.F., 2002. A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms. IEEE Transactions on Fuzzy Systems, 10(2), pp.155-170.

[12] Thaseen Ikram, S., Mohanraj, V., Ramachandran, S. and Balakrishnan, A., 2023. An intelligent waste management application using IoT and a genetic algorithm–fuzzy inference system. Applied Sciences, 13(6), p.3943.

[13] Nazarahari, M., Khanmirza, E. and Doostie, S., 2019. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. Expert Systems with Applications, 115, pp.106-120.

[14] Silva, L., Bellon, O.R.P. and Boyer, K.L., 2005. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. IEEE transactions on pattern analysis and machine intelligence, 27(5), pp.762-776.

[15] Suwarno, I., Cakan, A., Raharja, N.M., Baballe, M.A. and Mahmoud, M.S., 2023. Current trend in control of artificial intelligence for health robotic manipulator. Journal of Soft Computing Exploration, 4(1).

[16] Yin, X. and Pan, L., 2018. Enhancing trajectory tracking accuracy for industrial robot with robust adaptive control. Robotics and Computer-Integrated Manufacturing, 51, pp.97-102.

[17] Magkoutas, K., Rossato, L.N., Heim, M. and Daners, M.S., 2023. Genetic algorithm-based optimization framework for control parameters of ventricular assist devices. Biomedical Signal Processing and Control, 85, p.104788.

[18] Dang, X.K., Do, V.D. and Nguyen, X.P., 2020. Robust adaptive fuzzy control using genetic algorithm for dynamic positioning system. IEEE Access, 8, pp.222077-222092.

[19] Nonoyama, K., Liu, Z., Fujiwara, T., Alam, M.M. and Nishi, T., 2022. Energy-efficient robot configuration and motion planning using genetic algorithm and particle swarm optimization. Energies, 15(6), p.2074.

[20] Liu, X., Jiang, D., Tao, B., Jiang, G., Sun, Y., Kong, J., Tong, X., Zhao, G. and Chen, B., 2022. Genetic algorithm-based trajectory optimization for digital twin robots. Frontiers in Bioengineering and Biotechnology, 9, p.793782.

[21] Jamwal, P.K., Xie, S. and Aw, K.C., 2009. Kinematic design optimization of a parallel ankle rehabilitation robot using modified genetic algorithm. Robotics and Autonomous Systems, 57(10), pp.1018-1027.

[22] Chen, G., Hou, J., Dong, J., Li, Z., Gu, S., Zhang, B., Yu, J. and Knoll, A., 2020. Multiobjective scheduling strategy with genetic algorithm and time-enhanced A* planning for autonomous parking robotics in high-density unmanned parking lots. IEEE/ASME Transactions on Mechatronics, 26(3), pp.1547-1557.