JFSA

Journal of Futuristic Sciences and Applications
Vol. 3(1), Jan-Jun 2020, pp. 34-39
doi: 10.51976/jfsa.312006
www.gla.ac.in/journals/jfsa

Optimal Control of Manipulator Using Genetic Algorithm

Yogendra Kumar and Hemant Gupta***

ABSTRACT

(GA) is a Metaheuristic-based optimization method that addresses difficult optimization issues by simulating biological evolution and the survival of the fittest principle in natural contexts. The genetic algorithm (GA) method is presented and then assessed on a control issue to determine the optimal control structure for a certain time horizon. It is necessary to identify the various control parameters in order to execute the various control rules. This is ambiguous since there is no straightforward method for calculating these parameters for nonlinear systems. The introduction of the Genetic Algorithm, a metaheuristic optimization technique for determining the ideal nonlinear controller parameters, is our contribution. Using a dynamic model of a two-link rigid robot manipulator, the obtained results support the recommended optimal control strategy for the regulation and trajectory tracking problem, which is based on intelligent control and GA.

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 applications. Repeatability is a measure of a robot's ability to repeatedly return to the same posture. The capacity of a robot to move properly to a given pose in three-dimensional space is defined as accuracy. Robot accuracy is attributed to robot links and joint angle, but the statistical significance of robot parameter tolerances and relationships between these parameters has not yet been investigated. Although some academics have sought to investigate the impact of different robot parameters on performance, the development of a simulation approach to investigate robot parameter tolerance is unusual [1-6]. The optimal tolerance design technique requires consideration of the link size, joint angles, and torque delivered by each robot part to achieve the required precision in robot performance. 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.

**Corresponding author; Assistant Professor, Department of Electrical Engineering, GLA University, Mathura, Uttar Pradesh, India (E-mail: yogendra.ee@gla.ac.in)*
***Assistant Professor, Department of Electrical Engineering GLA University, Mathura, Uttar Pradesh, India. (E-mail: hemant.gupta@gla.ac.in)*

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. 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 can be described mathematically in a variety of ways. The problem of kinematics is to characterize the motion of the manipulator without taking forces and torques into account. These equations determine the location and orientation of the end effector given the joint variable values and the joint variable values given the end effector position and orientation. Dynamic modelling necessitates the development of equations that clearly explain the relationship between force and motion.When modelling robot motion and building control systems, these equations are critical. The framework is intended for manipulators with rigid links and solely revolute joints, although it can be adapted 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.

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

Complex manipulators with multiple degrees of freedom will produce enormous dynamic systems, and it may be advantageous to assess subsystems independently. Each element of the inertia matrix, as well as the gravity and Coriolis components, is evaluated separately at the end of the framework. The Matlab conversion code used at the end may be applied to any expression and is a simple way to convert Maple code to Matlab code. It should be noted that the Matlab conversion algorithm does not support joint variable derivatives. These values must be changed in Matlab as desired. Denying symbolic values is optional if the goal is a less detailed model, but keep in mind that the more symbolic values and zeros denied, the larger the dynamic model. The final equations that have been obtained by the dynamic model are-

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

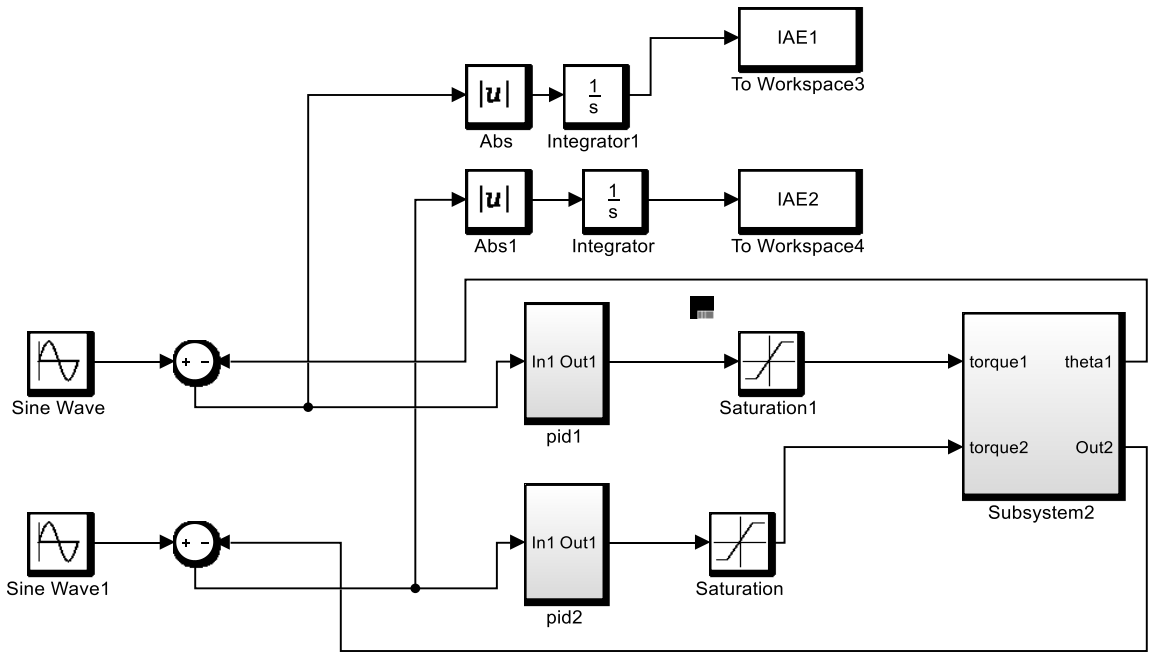
Journal of Futuristic Sciences and Applications, Volume 3, Issue 1, Jan-Jun 2020
Doi: 10.51976/jfsa.312006

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

3.0 Optimization Through Genetic Algorithm

GA was inspired by Darwin's theory of evolution, which simulates the survival of fitter organisms and their genes. The GA algorithm is a population-based algorithm. Each solution represents a chromosome, and each parameter represents a gene. GA uses a fitness function to assess the fitness of every individual in the population. With the purpose of improving defective solutions, the best responses are chosen at random using a selection mechanism). This operator is more likely to select the best solutions because probability is proportional to. The possibility of choosing suboptimal solutions enhances the likelihood of avoiding local optima. This suggests that good solutions can be extracted from a local solution utilizing other solutions. Crossing individuals leads to the exploitation of the 'region' between the two parent solutions. Mutation helps this method as well. This operator alters the genes on the chromosomes at random, preserving population diversity and enhancing GA's inquisitiveness. The mutation operator, like nature, may result in a significantly different outcome. Better solution, which will result in the global optimum of other solutions The GA method begins with a randomly generated population. 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

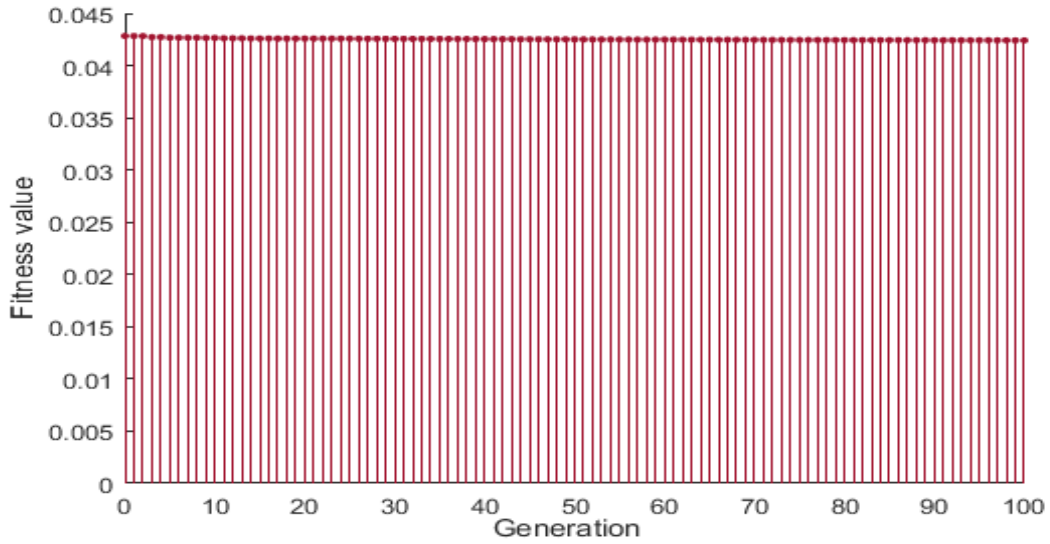


The figure above shows the optimized controller on which GA has been applied, the dynamic model described above is being treated here as a plant on which a controller is attached here Integral absolute error has been chosen as an objective function. The GA algorithm starts with a population of people 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

The results are achieved by using the optimal control settings of each connection angle after 120 iterations of the algorithm with a population size of 100. As shown in these figures, for the obtained parameters, the responses of both links have tracked the desired trajectories without overshoot, despite the fact that their initial conditions differ from the tracked trajectories' initial points, and both errors converge to zero in a short period of time. It was demonstrated that GAs can efficiently locate robot settings and produce results. Changes in starting state are also observed to have an effect on performance variations, but other dynamic variables have no effect. The plots are depicted 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.

Doi: 10.51976/jfsa.312006

5.0 Conclusion

The genetic algorithm-based robot dynamics control with the least amount of error has been tested. It was proved that GAs is capable of efficiently locating robot settings as well as producing results with the highest accuracy and least mistake. It is conceivable because the method only requires position feedback and final equations of the two links depicted in the dynamic model and does not involve measuring the speed and acceleration of the links, which is sometimes only possible through finite difference computations, resulting in significant identification errors. The operation of a genetic algorithm might be time-consuming, which is considered a disadvantage of the algorithm. However, huge advances in computer technology will significantly alleviate this disadvantage in the near future. Future study could be conducted in the direction of a cost connected to the objective function the model takes into account both deterministic and random flaws.

References

1. G. Silva, P. Costa, L. Rocha, and J. Lima, "Path planning optimization for a mobile manipulator." in Central European Symposium On Thermophysics 2019, 2019.

2. M. H. Korayem, M. Irani, A. H. Charesaz, A. Korayem, and A. Hashemi, "Trajectory planning of mobile manipulators using dynamic programming approach," Robotica, vol. 31, no. 4, pp. 643–656, 2012.

3. R. Gill, D. Kuli, and C. Nielsen, "Spline path following for redundant mechanical systems." in IEEE Transactions on Robotics, 2015, pp. 1378–1392.

4. T. Pardi, V. Maddali, V. Ortenzi, R. Stolkin, and N. Marturi, "Path planning for mobile manipulator robots under non-holonomic and task constraints," in Proceedings of The 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2020). IEEE, 2020.

5. C. Hu, W. Chen, J. Wang, and H. Wang, "Optimal path planning for mobile manipulator based on manipulability and localizability," in Proceedings of The 2016 IEEE International Conference on Real-time Computing and Robotics(RCAR), 2016. IEEE, 2016.

6. R. Gill, D. Kuli, and C. Nielsen, "Path following for mobile manipulators," in Springer Proceedings in Advanced Robotics Robotics Research, 2017, pp. 527–544.

7. A. Akhtar, C. Nielsen, and S. L. Waslander, "Path following using dynamic transverse feedback linearization for car-like robots," IEEE Transactions on Robotics, vol. 31, no. 2, pp. 269–279, 2015.

8. R. Gill, D. Kuli, and C. Nielsen, "Robust path following for robot manipulators," IEEE Transactions on Robotics, 2013.

9. D. Youakim and P. Ridao, "Motion planning survey for autonomous mobile manipulators underwater manipulator case study." in Robotics and Autonomous Systems, 2018, pp. 20–44.

10. M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," Computer, vol. 27, no. 6, pp. 17–26, 1994