A genetic algorithm approach for parameter optimization of a 7DOF robotic manipulator

C. West*, A. Montazeri*, S. Monk*, C.J. Taylor*

**Engineering Department, Lancaster University, Gillow Avenue, Lancaster, England, LA14YW*
(Tel: +44 (0) 1524 593660, e-mail: a.montazeri@lancaster.ac.uk)

Abstract: In this paper the problem of dynamic modeling and parameter estimation of a seven degree of freedom hydraulic manipulator is investigated. The numerical model is developed in Simulink using SimMechanic and Simscape toolboxes with unknown/uncertain parameters. The aim of this paper is to develop a mechanism that enables us to find a feasible set of parameters for the robot that is consistent with measurements of the input, output, and states of the system under noisy and unknown operating conditions. As the first step a genetic algorithm is developed to solve an output error system identification problem for a specific joint, i.e. joint 2, such that the parameters of the joint converge to the desired set of parameters within an acceptable accuracy. The results can be straightforwardly extended to all joints of the manipulator.

Keywords: Parameter estimation, System identification, Nonlinear model, Genetic algorithm, Mathematical modeling

1. INTRODUCTION

Industrial robots are proven to be an invaluable asset to take the place of human beings in many difficult and hazardous situations, such as manufacturing (Zhang et. al., 2014) and nuclear decommissioning (Bogue, 2011). Working in semi-structured, unstructured, dynamic, and harsh environments necessitates emergence of smarter, faster, and cheaper industrial robots with the capability of showing human traits such as sensing, dexterity, memory and trainability. As a result of progress in this area, nowadays industrial robots are taking on more complicated jobs such as picking and packaging, testing or inspecting products, cutting, and welding, which are all common problems with particular importance for automation in both manufacturing and decommissioning applications. Considering these facts, two Hydrolek hydraulically actuated manipulators, each with 7DOF (i.e. six rotary joints and one gripper) have been attached to a Brokk-40 mobile platform and developed at Lancaster University for R&D in relation to tasks such as the decommissioning, repair and maintenance of nuclear plants (see e.g. Bakari et. al. 2007; Taylor and Robertson, 2013).

[illegible] compensate the dynamics of this manipulator requires the development of techniques to capture the dynamic behaviour of the system accurately, and to estimate parameters of the developed model under different operating conditions. The importance of this problem in the robotic context is investigated in (Swevers et al., 1996). To address this nonlinear and non-convex problem, in this article a Genetic Algorithm (GA) is utilised. GAs are powerful global search optimization algorithms and have been used to solve various problems in control and system identification (Nguyen et al., 2014, Kristinsson and Dumont, 1992, Chang, 2007) including robotics (Yanrong and Yang, 2004, Nearchou, 1998) and manufacturing (Mak et al., 2000, Dingwei et al., 2001). From a control perspective, for example in the article by (Kwok and Sheng, 1994) a genetic algorithm is used to tune the parameters of a PID control system on a 6-DOF robot arm, a Puma560. Their results show that the genetic algorithm approach is better for parameter optimisation than traditional methods such as trial and error and empirical approaches. Similarly (Šitum and Ciković, 2014) focus on tuning PID gains, however this time for a hydraulic manipulator and focussing on one joint rather than the whole arm. The GA is used as part of the control system, generating PID gains to control the actuator. This allows better control during the non-linear behaviour of the hydraulic actuator, where traditional methods approximate the system as linear.

From a system identification point of view, genetic algorithms are used by (Grecu et al., 2009, Jafari et al., 2007) for tuning the model parameters of a robot arm at the design stage. They use a 3-DOF serial manipulator powered by electric motors and gearboxes as an example. The optimisation is on the weight of the gearbox and link lengths, with the aim of minimising manufacturing cost whilst [illegible]a. The main difference [illegible] Grecu et al., 2009, Jafari et al., 2007), and what is done in the present article, is that the former were modelling at the design stage whereas here an existing arm with unknown or time-varying parameters due to age and use is modelled. Hence, rather than trying to optimise the manipulator design, it is attempted to match the measured system performance. The fundamental reason for utilizing a GA as a nonlinear optimisation tool, is that we already have a potentially useful nonlinear mechanistic model for the arm but it has unknown parameters.

Another area in which GAs have found their way into robotics is for path planning. For instance, (Albert et al.,

2009) used a GA for path planning of a 3-DOF planar manipulator. Given a current and target position, the GA was used to find a path that minimises joint angle change whilst avoiding objects. Another area of robotics where GAs are used is for solving the inverse kinematics problem (Nearchou, 1998). This shows the range of tasks in which a GA can be used within robotics.

Nonetheless, there appears to be relatively few articles looking at applying GAs to hydraulic robot manipulators. One paper that does look at hydraulic manipulators is (Rouvinen and Handroos, 1997), where a 3-DOF log crane with a reach of over six and a half meters is investigated. The aim of this paper was to use GAs in conjunction with neural networks to compensate for the deflection of the links, and to improve accuracy in reaching a target position. Another paper that investigates hydraulic manipulators is (Šitum and Ciković, 2014) mentioned above, where a genetic algorithm was used to tune the PID gains to control a single hydraulic cylinder. GAs were used to tune the gains due to the difficulty in modelling the non-linear electro hydraulic system.

The aim of present research is to use a GA to estimate parameter values for a hydraulic manipulator, i.e. the HydroLek manipulator, to support the development of control systems for nuclear decommissioning and manufacturing applications. The main problem in working with this robot is that the actual values of the parameters of the manipulator are either unknown or may have changed through time from when the manipulator was first developed. The main contribution of the paper is design of a genetic algorithm with the capability of estimating its parameters using an output error system identification framework for the developed mechanistic model (Montazeri and Udo, 2016) of the HydroLek arm. It is expected to have not only a close match between the output response of the model and that of the real arm but also to achieve a set of parameters which are close to their ‘true’ values.

2. DYNAMIC MODEL OF THE MANIPULATOR

Different subsystems of the 7-DOF hydraulically actuated tele-operated robotic manipulator, i.e. the Hydrolek arm, are shown in the schematic block diagram of Fig. 1. Due to complexity in dynamic modelling of such a non-linear system, the aim is to develop a reliable simulator by which the dynamic and kinematic characteristics of the arm can be derived. This is essential for the design and implementation of different joint-level and supervisory control algorithms to accomplish difficult decommissioning or manufacturing tasks, such as remote pick and place, cutting, and welding autonomously. The approach adopted here is to investigate the modelling of various components of the manipulator using numerical and experimental techniques, and then to integrate them in a unified Simulink model for the purpose of system identification, parameter estimation, and the design of end effector trajectories, as well as the wider control objectives. This process is usually referred to as Robot Calibration in robotic terminology. The model is developed

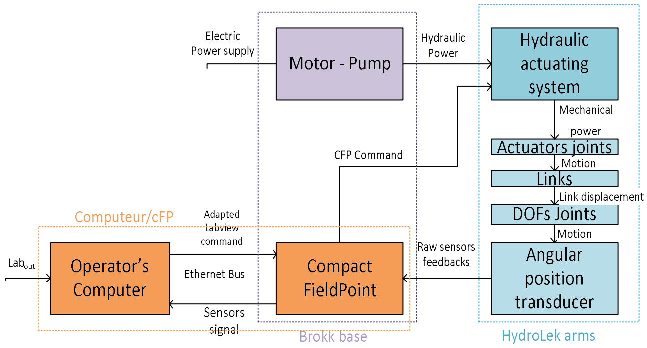

Fig. 1. High level description of 7-DOF hydraulically actuated tele-operated robot manipulator (Montazeri and Ekotuyo, 2016).

for the MATLAB Simulink environment and, in the final version of the model, will comprise all the mechanical, electrical, and hydraulic parameters of the manipulator. Details of the modelling equations for each block in Fig. 1 are explained in (Montazeri and Ekotuyo, 2016, and Antoine, 2014) but are omitted here for brevity and because the focus of the present article is on the implementation of the GA for parameter estimation. Unfortunately, due to the age of the arms, data sheets containing values for many parameters are unavailable, and certain parameters may have changed over time. Because of this it is intended to develop a mechanism using GAs that enables us to estimate and update certain parameter values of the model any time they are needed. As the starting point the focus is on one joint, with concentration on the mechanical parameters of the arm. Lacking data sheets stating the mass of the individual links, the overall mass of the arm and some guess work was used to define the starting values of link masses.

The same problem applies for the spring stiffness and damping coefficient of the joint. By subsequently using the estimated mass for each link, the inertia can be calculated based on the geometry of the link, which is known from the CAD model of the arm. All of these nine parameters are included in the process of the GA to be estimated with sufficient accuracy. The proposed genetic algorithm is developed and explained in the next section.

3. THE PROPOSED GENETIC ALGORITHM

GA is a heuristic global optimisation method based on the biological principal of natural selection, where the strongest individuals will survive and reproduce. In the GA the individual parameters are encoded as strings of numbers called chromosomes, so for example one chromosome will contain a value for each of the parameters being investigated. The process starts by creating a random population of potential solutions; these are then evaluated using a fitness function. A weighted roulette wheel selection method is then used to find the strongest members of that population to pass them through to the next stage, crossover. The crossover stage selects two random parent chromosomes and combines them to form two child chromosomes, for example, by combining the front half of one parent with the end half of the other parent. Mutation is the final stage, where single elements may be randomly swapped to create a more diverse population. The process then starts again with the new population, and repeats until either a minimum fit is reached or a specific number of iterations are passed. Figure 2 (Montazeri et. al. 2003, Montazeri and Poshtan 2009) shows the flow chart of the GA adopted for the present optimization purpose. In the sequel, each block in Fig. 1 will be explained and expanded in more detail, to tailor the algorithm to the specific problem at hand. Two different coding schemes are investigated, integer and multi variable binary string (MVBS), to find which gives the best performance in converging to a global minimum solution. Using either coding scheme, the chromosomes themselves look the same, as shown in (1),

$$\text{Chromosome} = [K\,D\,S\,M_1\;M_2\;M_3\;M_4\;M_5\;M_6] \qquad (1)$$

where K is the gain representing the whole block of the electro hydraulic actuator, D is the joint damping coefficient, S is the joint spring stiffness and M_1 to M_6 are the link masses. In the MVBS scheme, each element of the chromosome is represented by a 16 bit binary number, whereas in the integer coding scheme each element is represented by an integer.

The next step is to integrate the GA code with the Simulink model of the system. For each iteration of the algorithm, the Simulink model is run with the parameters in that chromosome and the output is compared to measured experimental data using a fitness function, to generate a value of fit for that chromosome. This fit value is used to assess the strength of the chromosome. The fitness value is determined by comparing the result of running the Simulink model with each chromosome of parameters in that iteration against experimental data for the joint. The closer the simulation output is to the experimental data the smaller the fit value.

As will be investigated later, selection of the fitness function plays an important role in the convergence behaviour of the proposed GA. A particular form of the fitness function converts the GA to a suitable algorithm for the estimation of parameters of a nonlinear model, while other selection of fitness function ensures the GA performs well for output error system identification purposes. Six different fitness functions were evaluated to find the one with the best performance, three taking the form of (2) and three as shown by equation (3):

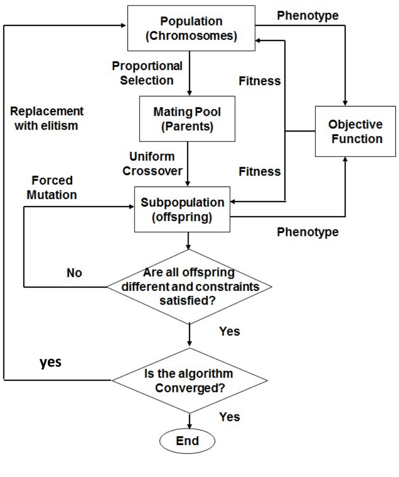

Fig. 2. Flowchart showing genetic algorithm process.

$$Fit = -\frac{norm(e,\alpha)}{norm(Experimental\ data,\alpha)} \qquad (2)$$

$$Fit = -norm(e,\alpha) \qquad (3)$$

where the error is generated by comparing the simulation and experimental data as follows:

$$e = Simulation\ data - Experimental\ data$$

Here α defines the norm type and takes the following values:

$$\begin{aligned} if\ \alpha &= 1\ One\ norm\ or\ Taxicab\ norm \\ \alpha &= 2\ Euclidian\ norm \\ \alpha &= Infinity/\max norm \end{aligned}$$

Both the experimental data and the simulation are sampled at 0.01 second intervals, and each point of the simulation data is compared against the corresponding point in the simulating data to find the difference between the two.

4. RESULTS

4.1 Training phase of the algorithm

Further to the fitness function and the coding scheme explained in the previous section, the performance of the genetic algorithm in finding the best estimation for parameters of the developed model is heavily influenced by several other factors, namely, crossover rate, crossover type, mutation rate, and population length. With the aim of finding the best estimate of parameters for the developed nonlinear model, it is necessary to train the genetic algorithm and find the best possible parameters for this specific problem. For this purpose a specific set of parameters shown in the first row of Table 1 is considered for the Simulink model to generate the simulation data suitable for this part. The reason for this is that it allows us to look at the estimated value of the parameters outputted by the GA and see how close they are to the real values already set in the Simulink model. It also makes clear under which circumstances the algorithm has the capability to converge to the real values of the parameter, rather than getting stuck in a different global minimum point. This happens when the fitness value would be low but the estimated parameters are a long way off from the value of the real parameters. Initially, the algorithm is run by just looking at the gain, spring stiffness, damping coefficients, and mass for each link. The inertia of each link is calculated by the SimMechanics blocks that model the individual link. The parameters of the model are initialized to randomly chosen values, as shown in the second row of Table 1.

Table 1 – Initial versus true value of parameters of the model.

Parameters	Gain	Damping	Stiffness	Mass of link (kg)					
				1	2	3	4	5	6
Tru.	0.2	231.8	53.4	4.6	5.1	22.5	1.6	4.1	4.7
Init.	0.1	100	20	0.6	1.3	2.8	1.2	2.2	1.7

4.2 Evaluation of different coding schemes
Within the GA, the chromosomes are coded in some way before being passed to crossover and mutation operations, then decoded later to get back to the actual parameter values. The coding takes the GA between the two spaces it works in: the genotype coding space and the phenotype solution space. In the phenotype space each chromosome contains a value for each parameter within that chromosome, in genotype space each chromosome is represented in some way depending on the coding scheme.
Here, the performance of the algorithm for integer and multi variable binary coding schemes is investigated. As can be seen from Fig. 3 the MVBS coding scheme produces more varied results with the capability to converge to a better value of the fitness function for the developed model. Therefore this coding scheme will be used later to solve the practical problem.

4.3 Evaluation of different crossover types
In development of the algorithm, two different methods of performing crossover are investigated: pointwise and uniform crossover. For the pointwise crossover scheme, chromosomes are broken into several segments and each iteration two parent chromosomes swap segments between the segmentation points. Uniform crossover works by swapping every other gene of one of the parent chromosomes with the other parent, so each child ends up with 50 percent of each parent chromosome. The results of running the GA for these two different crossover types are compared in Fig. 4. In execution of the algorithm all other parameters are set according to Table 4. As shown by Fig. 4, the difference between the two methods for estimation of parameters of the model is considerable, with the pointwise method being much better for this problem.

4.4 Evaluation of different crossover rates
The result of running the GA for different crossover values while the other parameters are set according to Table 4 is shown in Fig. 5. As can be seen from Fig. 5, the crossover rate has considerable effect on the speed of convergence of the genetic algorithm. A crossover value of 1 means that at every iteration all the parent chromosomes will create new child chromosomes, so every iteration will contain different chromosomes to the one before. Crossover values of 0.6 and 0.8 take almost the same time to converge and, despite 0.8 having a higher initial value of fitness, it converges to the lowest value, suggesting that a crossover value of 0.8 gives the best performance in this particular case. In fact, with this crossover value, the best compromise is to keep a balance between the diversity of populations and the need to force the output to determine the fittest individuals.

4.5 Evaluation of different fitness functions
One of the most important factors in the development of any GA for the specific problem at hand, is selection of the suitable fitness function to discriminate the individuals properly. Different fitness functions considered here are given in equations (2) and (3). The results of running the GA for these fitness functions are shown in Fig. 6, while Fig. 7 shows a zoomed in section of Fig. 6 and shows more clearly the number of iterations the GA requires converging for each fitness function. Table 2 compares the relative error of the estimated parameters for each fitness function. The sum of these errors listed in the third column of Table 3 is considered as a measure for determining the suitability of a specific fitness function for parameter estimation. Table 3 shows the parameter values corresponding to the plots in Fig. 6. This result confirms that convergence of the estimated parameters to the true values needs particular attention in the development of the GA algorithm. Because different fitness functions result in different numerical outputs, it is not possible to compare the learning curves of the GA for different fitness functions in one plot. To solve this problem, all the learning curves are normalized between 0 and 1. The results plotted in Fig. 7 show that norm two (the Euclidian norm) reaches the final fitness value the soonest, with the infinity norm (the max norm) taking the longest to reach its final value.

Table 2. Relative percentage error of estimated parameters for each fitness function.

Fitness Function	Gain	Damping	Stiffness	Mass of link (kg)					
				1	2	3	4	5	6
1	50	26.0	42.5	19.6	5.9	43.1	12.5	22.0	106.4
2	100	82.6	43.1	84.8	94.1	96.4	250.0	146.3	72.3
3	50	57.9	29.4	23.9	9.8	84.0	93.8	19.5	21.3
4	50	32.5	47.6	23.9	80.4	78.2	393.8	24.4	12.8

Key: Fit as in equation (3) 1.Norm inf. (no den) 2. Norm one 3. Norm two. Fit as in equation (2) 4. Norm inf. with denominator.

Table 3. Comparison of output and parameters error for each fitness function shown on the plot in Fig. 6.

Fitness Function	Lowest fitness value	Output error index	Parameters error index
Norm inf no den.	-0.6458	0.645	327
Norm inf with den.	-0.1275	0.651	743
Norm one no den.	-836.54	0.85	969
Norm two no den.	-14.5172	0.82	389

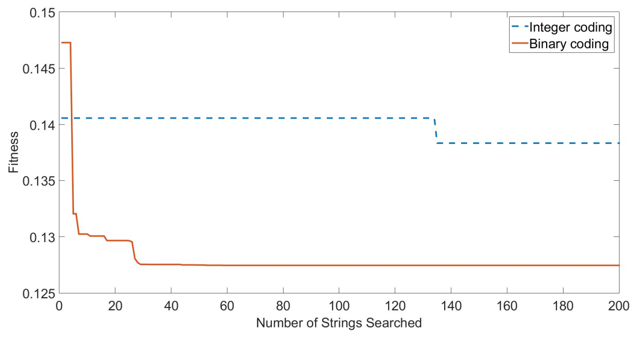

Fig. 3. Comparison of binary and integer coding schemes.

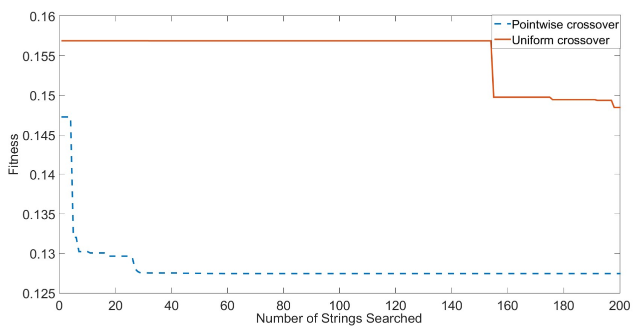

Fig. 4. Comparison of pointwise and uniform crossover methods.

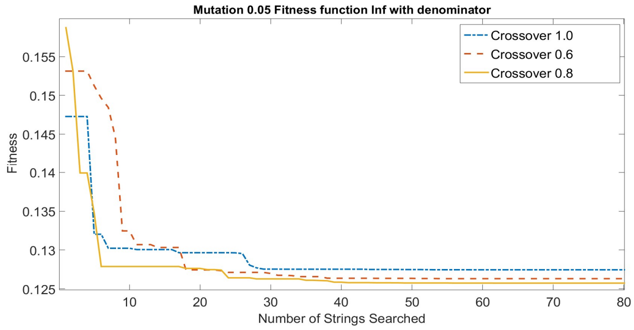

Fig. 5. Plot showing the effect of different crossover rates on the elite fitness value for 200 iterations.

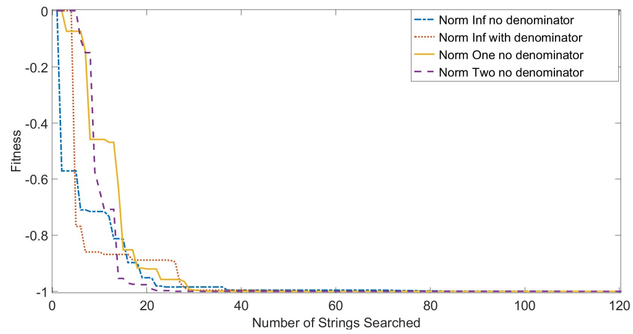

Fig 6. Plot showing the effect of different fitness functions on the elite fitness value.

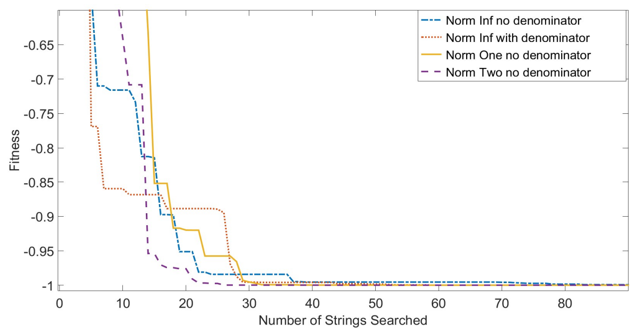

Fig 7. Zoomed in section of Fig. 6 comparing the convergence rates of the algorithms.

The first column of Table 3 compares the absolute value of the fitness functions after convergence of the GAs. Although norm two has the fastest convergence rate, it is evident from this column that it has a relatively high fitness value, whereas the infinity norm with the slowest convergence rate has a very low fitness value. The capability of different fitness functions to find the best solution, in terms reaching the minimum output identification error and parameter estimation error, are compared in the second and third columns. The second column of Table 3 compares the infinity norm of the output identification error for each norm, while third column lists the sum of relative estimation errors for different norms. As can be seen from Table 3, the infinity norm without denominator results in both the lowest output identification error and the parameters estimation error suggesting that it performs the best in this case.

4.6 Final tune of the proposed algorithm

From the results achieved in the previous sections, it can be concluded that using the values in Table 4 should give the best performance for the GA. Figure 8 shows the final output after running the GA with these conditions. The proposed GA adapts the parameters of the model in an output error identification framework, while keeping the estimated model parameters close to the true parameters chosen for the model. As the plot in Fig. 8 shows, the estimated model output tracks the simulation output reasonably well, but tends to overshoot at the peaks. As can be seen from Table 5 and Fig. 8, choosing the fitness function as the relative infinity norm of the error, leads to an algorithm with the capability to estimate the parameters of the model in a reasonable proximity to the parameters used to generate the simulation data, in comparison to the other fitness functions. However, further work is required to improve the estimation accuracy of the proposed algorithm.

Table 4. The final value of the parameters used for the proposed GA.

Parameter	**Value**
Coding scheme	Multivariable binary coding
Crossover rate (Pc)	0.8
Mutation rate (Pm)	0.05
Parent selection	proportional
Crossover type	pointwise
Population size	70
Fitness function	infinity norm

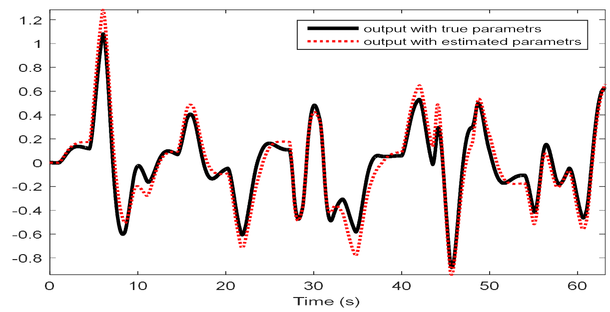

Fig 8. Comparison of the output of the dynamic model for true and estimated parameters.

Table 5. Estimated output versus true parameter values of the model.

Parameter	Gain	Damping	Stiffness	Mass of link (kg)					
				1	2	3	4	5	6
Est.	0.03	93	99.9	9.2	20	5.9	9.6	15.54	1.2
Tru	0.02	48	87.4	5.8	12.7	6.9	7.75	8.31	4.2

5. CONCLUSIONS

In this paper, the problem of dynamic modelling and parameter estimation of a 7-DOF robot manipulator, i.e. the HydroLek arm, is investigated. Due to the complexity of the dynamic behaviour of the system, a mechanistic model using Simulink is developed. However, the parameters of the model are subject to change because of device aging and time-varying characteristics of different operating conditions. To overcome this problem for the purpose of future simulation and controller design work, this article has exploited and refined a genetic algorithm to estimate the parameters of the model using an output identification framework. This is accomplished in two steps. In the first step, the parameters of the GA, i.e. coding scheme, crossover type, crossover rate, mutation rate and fitness function, are all tuned on the basis of simulation data. The results show that the developed GA has the capability to estimate the parameters of the dynamic model with a reasonable accuracy and the output of the model follows the simulated output closely. In the second step, the proposed GA is utilized to estimate the parameters of the model based on measured experimental data. The developed model will be used for simulation and model based controller design, and this will be reported in future articles.

REFERENCES

Albert, F., Koh, S., Chen, C., Tiong, S., Edwin, S., (2009). Optimizing Joint Angles of Robotic Manipulator using genetic Algorithm. *Proceedings of International Conference on Computer Engineering and Applications (ICCEA 2009)*, pp. 134-139, Singapore.

Antonie, M. (2014). *Dynamic modelling and parameter estimation of a hydraulic actuated manipulator.* Master thesis, Lancaster University.

Bakari, M.J., Zeid, K.M., Seward, D.W. (2007). Development of a Multi-arm mobile Robot for Nuclear Decommissioning Tasks. *International Journal of Advanced Robotics Systems*, vol. 4, pp. 384-406.

Bogue, R. (2011). Robots in the Nuclear Industry: a Review of Technologies and Applications, *International Journal of Industrial Robot*, vol. 38, pp. 113-118.

Chang, W.-D. (2007). Nonlinear system identification and control using a real-coded genetic algorithm. *Applied Mathematical Modelling,* vol. 31**,** pp. 541-550.

Dingwei, W., Yung, K. L., IP, W. H. (2001). A heuristic genetic algorithm for subcontractor selection in a global manufacturing environment. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 31, pp. 189-198.

Grecu, V., Grecu, L., Demian, G., Demian, M., Mastorakis, N., Martin, O. (2009). A Genetic Algorithm for Optimization in Conceptual Robot Manipulator. *WSEAS International Conference. On Mathematics and Computers in Science and Engineering.*

Jafari, A., Safavi, M., Fadaei, A. A. (2007). Genetic Algorithm to Optimum Dynamic Performance of Industrial Robots in the Conceptual Design Phase. IEEE 10th International Conference on Rehabilitation Robotics (ICORR 2007), pp. 1129-1135.

Kristinssion, K., Dumont, G. (1992). System identification and control using genetic algorithms. *, IEEE Transactions on Systems, Man and Cybernetics,* vol. 22**,** pp. 1033-1046.

Kwok, D. P., Sheng, F., (1994). Genetic algorithm and simulated annealing for optimal robot arm PID control. *IEEE World Congress on Computational Intelligence and Evolutionary Computation*, vol. 2, pp. 707-713.

Mak, K. L., Wong, Y. S., Wang, X. X. (2000). An Adaptive Genetic Algorithm for Manufacturing Cell Formation. *The International Journal of Advanced Manufacturing Technology*, vol. 16, pp. 491-497.

Montazeri A., Udo, E. J. (2016). Development of Dynamic Model of a 7DOF Hydraulically Actuated Tele-Operated Robot for Decommissioning Applications. To appear *in Proceeding of IEEE 2016 American Control Conference,* Boston, MA, USA.

Montazeri, A., Poshtan, J., Kahaei, M.H., (2003). Optimal placement of loudspeakers and microphones in an enclosure using genetic algorithm. *IEEE Conference on Control Applications*, pp 135-139.

Montazeri, A., Poshtan, J. (2009). GA-based optimization of a MIMO ANC system considering coupling of secondary sources in a telephone kiosk. *Applied Acoustics,* vol. 70**,** pp. 945-953.

Nearchou, A.C. (1998). Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm. *Mechanism and Machine Theory*, vol. 33, pp. 273-292.

Nguyen, A.-T., Reiter, S., Rigo, P. (2014). A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy,* vol. 113**,** pp. 1043-1058.

Rouvinen, A., Handroos, H. (1997). Robot positioning of a flexible hydraulic manipulator utilizing genetic algorithm and neural networks. *4th Annual Conference on Mechatronics and Machine Vision*, pp. 182-187.

Situm, Z., Cikovic, Z. (2014). Optimization of control parameters for servo hydraulic systems using genetic algorithms. *Hidravlika–Regulacija*, vol. 3, pp. 198-203.

Swevers, J., Ganseman, C., De Schutter, J., Van Brussel, H. (1996). Experimental Robot Identification Using Optimised Periodic Trajectories. *Mechanical Systems and Signal Processing*, vol. 10, pp. 561-577.

Taylor, C.J. and Robertson, D. (2013). State-dependent control of a hydraulically-actuated nuclear decommissioning robot, *Control Engineering Practice*, 21, 12, pp.1716-1725.

Yanrong, H., Yang, S.X. (2004). A knowledge based genetic algorithm for path planning of a mobile robot. *Proceedings on IEEE International Conference on Robotics and Automation*, vol. 5, pp. 4350-4355.

Zhang, G.Q., Li, X., Boca, R., Newkirk, J., Zhang, B., Fuhlbrigge, T.A. (2014). Use of Industrial Robots in Additive Manufacturing– A Survey and Feasibility Study, 41*st* *International Symposium on Robotics,* pp. 1-6, Munich, Germany.