



Dynamic modeling and parameter estimation of a hydraulic robot manipulator using a multi-objective genetic algorithm

A. Montazeri, C. West, S. D. Monk & C. J. Taylor

To cite this article: A. Montazeri, C. West, S. D. Monk & C. J. Taylor (2016): Dynamic modeling and parameter estimation of a hydraulic robot manipulator using a multi-objective genetic algorithm, *International Journal of Control*, DOI: [10.1080/00207179.2016.1230231](https://doi.org/10.1080/00207179.2016.1230231)

To link to this article: <http://dx.doi.org/10.1080/00207179.2016.1230231>



Accepted author version posted online: 30 Aug 2016.
Published online: 30 Aug 2016.



Submit your article to this journal [↗](#)



Article views: 21



View related articles [↗](#)



View Crossmark data [↗](#)

Publisher: Taylor & Francis
Journal: *International Journal of Control*
DOI: <http://dx.doi.org/10.1080/00207179.2016.1230231>

Dynamic modeling and parameter estimation of a hydraulic robot manipulator using a multi-objective genetic algorithm

A. Montazeri^{1,*}, C. West^{*}, S. D. Monk^{*}, C. J. Taylor^{*}

^{*}Engineering Department, Lancaster University, Gillow Avenue, Lancaster, England, LA14YW

(¹Corresponding author: Tel: +44 (0) 1524 593660, e-mail: a.montazeri@lancaster.ac.uk)

This article concerns the problem of dynamic modeling and parameter estimation for a seven degree of freedom hydraulic manipulator. The laboratory example is a dual-manipulator mobile robotic platform used for research into nuclear decommissioning. In contrast to earlier control model orientated research using the same machine, the article develops a nonlinear, *mechanistic simulation* model that can subsequently be used to investigate physically meaningful disturbances. The second contribution is to optimize the parameters of the new model, i.e. to determine reliable estimates of the physical parameters of a complex robotic arm which are not known in advance. To address the nonlinear and non-convex nature of the problem, the research relies on the multi-objectivization of an output error single performance index. The developed algorithm utilises a multi-objective Genetic Algorithm (GA) in order to find a proper solution. The performance of the model and the GA is evaluated using both simulated (i.e. with a known set of ‘true’ parameters) and experimental data. Both simulation and experimental results show that multi-objectivization has improved convergence of the estimated parameters compared to the single objective output error problem formulation. This is achieved by integrating the validation phase inside the algorithm implicitly and exploiting the inherent structure of the multi-objective GA for this specific system identification problem.

Keywords: Parameter estimation, System identification, Nonlinear model, Multi-objective genetic algorithm, Mathematical modeling

1. INTRODUCTION

Industrial robots have proven to be an invaluable asset to take the place of humans 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 the emergence of smarter, faster and cheaper industrial robots, sometimes with the capability of showing human traits such as sensing, dexterity, memory and trainability. As a result of progress in this area, industrial robots are now taking on more complicated jobs such as picking and packaging, testing or inspecting products, cutting and welding. These are all common problems with particular importance for automation in both manufacturing and nuclear decommissioning. Considering these facts, two Hydrolek hydraulically actuated manipulators, each with seven degrees of freedom (7-DOF, i.e. six rotary joints and one gripper: see Figure 1) have been attached to a Brokk-40 mobile platform and developed at Lancaster University for research into the decommissioning, repairs and maintenance of nuclear plants (see e.g. Bakari, Zied & Seward, 2007; Taylor & Seward, 2010; Taylor & Robertson, 2013).

Designing high performance control algorithms to compensate the dynamics of this manipulator requires the development of techniques to capture the dynamic behaviour of the system accurately, and to estimate the parameters of the developed model under different operating conditions. The importance of this problem in a robotic context is investigated by, for example, Swevers, Ganseman, De Schutter & Van Brussel (1996). In this regard, previous research for the Brokk-Hydrolek and other hydraulic manipulators (e.g. Taylor, Shaban, Stables & Ako, 2007; Taylor & Robertson, 2013) has demonstrated the practical utility of a quasi-linear, State Dependent Parameter (SDP) model. The parameters of the SDP model are functionally dependent on the measured variables (Young, 2011), such as joint angles and velocities in the case of manipulators. The approach is ideal for capturing the essential nonlinear behaviour of the system when a relatively straightforward dynamic equation is required for model-based control.

However, such SDP models are presently limited to individual manipulator joints, and they have not been designed to represent the complex mechanistic interactions between various mechanical, hydraulic and electrical components of the Brokk-Hydrolek system.

Instead of the SDP model structure, in this article we develop a nonlinear mechanistic model for the manipulators. In Section 2, this is accomplished by decomposing the system into several dynamical subsystems and subsequently integrating the different modules together. The model not only provides an accurate and detailed representation of such a complex mechatronic system, but also provides a grey-box model structure with physically meaningful parameters. In this manner, the research develops a reliable simulator by which the dynamic and kinematic characteristics of the manipulator can be analysed. This is essential for the design of joint-level and supervisory control algorithms to accomplish nuclear decommissioning or manufacturing tasks (such as remote pick and place, welding and pipe cutting). Some preliminary results in relation to modelling of the mechanical components are given by Montazeri & Udo (2016) but these authors do not consider the hydraulic model component or the parameter estimation problem. With regard to the latter, our approach in this article is to use a stochastic evolutionary technique based on a multi-objective Genetic Algorithm (GA).

Parameter estimation for robotic systems is a challenging task, especially when the number of degrees of freedom and the number of parameters to be estimated are high. The main difficulty is that the underlying problem is nonlinear and non-convex in nature. The most common technique to solve the non-convexity of such a problem is to use an equation error identification approach, which assumes that the dynamic model of the robot can be expressed in a linear form with respect to the unknown parameters, and hence the parameters can be estimated using Least Squares (LS) error minimisation criteria (Gautier & Poignet, 2001; Khalil & Dombre, 2002; Swevers, Verdonck & Schutter, 2007). The main problem with this technique relates to the presence of uncertainties due to modelling errors and measurement noise in the observation matrix, which results in bias and variance error in the estimated parameters. Various techniques are reported in the literature to help alleviate this problem. For example, band-pass filtering of joint positions provides one solution (Gautier, Janot & Vandanjon, 2013). Furthermore, Janot, Vandanjon & Gautier (2014) compare the performance of this technique with instrumental variables, total LS and

output error methods. More advanced robust estimation such as maximum likelihood and extended Kalman filter based techniques have also been proposed (e.g. Gautier & Poignet, 2001).

An alternative approach is to formulate an output error system identification problem in nonlinear LS terms and to use nonlinear programming (Gautier, Jubien & Janot, 2012; Gautier et al., 2013). When the performance function is smooth, a local optimization technique may then converge to the global optimum. However, Janot, Vandanjon & Gautier (2014) show that application of a multi-start Gauss-Newton algorithm for parameter estimation of a 6-DOF robot is not a proper strategy. The problem of multi-modality of the performance surface can be solved by the design of a suitable filter (Tohme *et al.*, 2007), and the technique is efficient when the first and second order derivatives of the performance surface are available analytically. Unfortunately, this assumption is not valid in the present case since, due the complexity of the model, no closed-form analytical equation exists. Furthermore, the performance indices assumed in the optimization process, such as the output error infinite norm, are not necessarily differentiable.

Therefore, a more effective approach to search a large parameter space is to use evolutionary-based algorithms. These include genetic algorithms (GA) (Affenzeller *et al.*, 2009), related multi-objective approaches (Coello, 2006; Zhou *et al.*, 2011), particle swarm optimization (PSO: Parsopoulos & Vrahatis, 2010) and the recently proposed differential evolution approach (Das & Suganthan, 2011). Gotmare, Bhattacharjee, Patidar & George (2016) discuss the suitability of GAs for various optimisation problems in system identification and filter design. Yao & Sethares (1994) and Nyarko & Scitovski (2004) provide similar in the context of parameter estimation more specifically. Use of PSO as an alternative to GA optimisation is primarily motivated by the relative simplicity of implementation and the ability to memorize the good solutions by all particles. As a result, PSO has been used for various control and identification problems (e.g. Montazeri, Poshtan & Yousefi-Koma, 2008; Ye, 2006; Sun & Liu,

2013). However, in contrast to the present parameter estimation problem, PSO is usually considered for real-time and adaptive applications because of its computational simplicity (Krusiński & Jenkins, 2005).

Vicente, Ayala & Coelho (2012) propose a multi-objective GA using the NSGA-II algorithm to tune the parameters of a PID controller for a 2-DOF robot arm. Simple GAs are also applied to tune PID parameters for more realistic applications, such as a 6-DOF Puma560 robot arm (Kwok & Sheng, 1994) and electro-hydraulic servo systems (Wu, Wang & Bai, 2012). In the specific case of hydraulic manipulators, Rouvinen & Handroos (1997) use GAs in conjunction with neural networks to compensate for the deflection of the links in a 3-DOF log crane. Jafari, Safavi & Fadaei (2007) use a GA to minimise manufacturing costs of 3-DOF serial manipulator at the *design stage*, rather than estimating the unknown physical parameters of an existing manipulator as in the present article. Al-Dabbagh *et al.* (2014) use GA optimisation to estimate the friction parameters of a single link 4-DOF surgery robot, while Bingül & Karahan (2011) use PSO to estimate the dynamical model parameters of the first three links of a Staubli RX-60 Robot. The latter two articles both make the assumption that the dynamic model of the robot is linear in the parameters. By contrast, the present research uses a multi-objective GA to estimate parameter values for the nonlinear simulation model of a 7-DOF hydraulic manipulator.

The main problem in working with the robotic platform shown in Figure 1, is that the actual values of the parameters are either unknown or may have changed through time from when the manipulator was first developed, due to age and use. By combining measured experimental data and the mechanistic models obtained in this article, a multi-objective GA is developed and used to estimate these parameters. The aim is not only a close match between the measured 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. Hence, in addition to the development of the mechanistic model, the second main contribution of the article is to investigate the use of GAs in a hydraulic manipulator context, and to compare the performance of a simple GA with a more complex, multi-objective approach, using both simulated and experimental data.

Sections 2 and 3 describe the mechanistic model and associated parameter estimation problem respectively. The two GA methods are introduced in section 4, in which improved convergence of the estimated parameters is achieved by exploiting the inherent structure of a multi-objective GA for this specific output error system identification problem. The article also includes an empirical evaluation of different crossover types, fitness functions and other settings, and develops recommendations for their particular implementation in the estimation of the mechanical and hydraulic subsystems. In this regard, the GA performance is considered in section 5, followed by the parameter estimation results in section 6 and the conclusions in section 7.

2. DYNAMIC MODEL OF THE MANIPULATOR

Each HydroLek manipulator has seven DOF, with a continuous (360 degree) jaw rotation mechanism and dual function gripper (Taylor & Robertson, 2013). In total, five sets of linear actuators and two rotary actuators are used (Figure 1). Each joint is fitted with potentiometer feedback sensors, allowing the position of the end-effector to be determined during operation. The end-effectors can also be equipped with a variety of tools, such as percussive breakers, hydraulic crushing jaws, excavating buckets and concrete milling heads. The joints are actuated via hydraulic pistons, which are powered via an auxiliary output from the hydraulic pump.

It is known that the dynamic equations of a manipulator can be derived using standard Newton-Euler theory as follows (Khalil & Dombre, 2002):

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{F}_c \text{sign}(\dot{\mathbf{q}}) + \mathbf{F}_v \dot{\mathbf{q}} = \boldsymbol{\tau} \quad (1)$$

where $\boldsymbol{\tau}$ denotes the vector of forces/torques generated by the hydraulic actuators, \mathbf{D} is the manipulator inertia matrix, \mathbf{C} is the coriolis and centrifugal matrix, \mathbf{g} is the gradient of the potential energy, \mathbf{F}_c is the coulomb friction and \mathbf{F}_v is the viscous friction. The vector \mathbf{q} contains all joint angles for the revolute joints. However, it would be extremely complex to analyse the dynamics of the HydroLek manipulator with different types of joints (such as revolute, cylindrical and gears) analytically in this way. Therefore,

use of equation (1) would impose several limitations on the derivation of a dynamic model, including a serial link assumption, infinitely rigid links and joints, and backlash free joints (Mavroidis *et al.*, 1998). Furthermore, for relatively high DOF, parametrisation of equation (1) becomes extremely complicated, even using symbolic software. To overcome these limitations, and to exploit existing Computer-Aided Design (CAD) models for the manipulator (see later), the present research instead relies on a numerical model to solve the equations of motion. This is achieved by modelling various components of the manipulator using numerical and experimental techniques, and subsequently integrating these into a unified simulation 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 the robotic literature (Calafiore & Indri, 2000).

The model comprises all elements of the system, as summarised in Figure 2, including the mechanical, electrical and hydraulic parameters of the manipulator. The command signal Lab_{out} is generated by a feedback controller or set directly by the operator, and is processed and calibrated before it can be sent to the National Instruments' Compact Field Point (CFP) module.

2.1. Compact Field Point Subsystem

The role of the computer in Figure 2 is to convert the signal Lab_{out} to a calibrated voltage command for the proportional amplifiers i.e. either V_{ampA_i} or V_{ampB_i} for joint i (depending on the sign of the Lab_{out} signal). Thus, for each joint i , the voltages V_{ampA_i} and V_{ampB_i} are expressed as follows:

$$V_{ampA_i} = \begin{cases} A_i + \left(|Lab_{out_i}| \cdot \frac{B_i - A_i}{100} \right) & \text{if } Lab_{out_i} > 0 \\ 0 & \text{if } Lab_{out_i} \leq 0 \end{cases}$$

$$V_{ampB_i} = \begin{cases} C_i + \left(|Lab_{out_i}| \cdot \frac{D_i - C_i}{100} \right) & \text{if } Lab_{out_i} < 0 \\ 0 & \text{if } Lab_{out_i} \geq 0 \end{cases}$$

(3)

In equations (2) and (3), $V_{amp_{A_i}}$ and $V_{amp_{B_i}}$ are the voltage input at the proportional amplifiers A and B respectively, while the parameters A_i, B_i, C_i, D_i are the joint i calibration coefficients and should be tuned (see section 3). The voltage inputs to the proportional amplifiers are shown in Figure 3.

2.2. Hydraulic Actuator Subsystem

The hydraulic actuating subsystem shown in Figure 3 consists of two further subsystems, namely the valves and the hydraulic cylinders. The valve system is decomposed into the proportional amplifier, solenoid valve and directional valve components. The four-way directional spool valve operates with the solenoid valve coil currents I_{coil_A} and I_{coil_B} . The voltage commands V_{amp_A} and V_{amp_B} applied to the proportional valves A and B are converted to the current coil, and energize the solenoids A and B:

$$I_{coil_{H_i}} = \begin{cases} I_{min_i} & \text{if } V_{amp_{H_i}} = 0 \text{ V} \\ V_{amp_{H_i}} \frac{I_{max_{H_i}} - I_{min_{H_i}}}{10} + I_{min_{H_i}} & \text{if } 0 < V_{amp_{H_i}} < 10 \text{ V} \\ I_{max_i} & \text{if } V_{amp_{H_i}} \geq 10 \text{ V} \end{cases} \quad (4)$$

where $V_{amp_{H_i}}$ is the joint i voltage signal at the input of proportional amplifier H (in which H is either A or B), $I_{coil_{H_i}}$ is the output amplifier H current that will energize the solenoid H of the joint i actuating system, and $I_{max_{H_i}}$ and $I_{min_{H_i}}$ are the saturation current parameters associated with the amplifiers H of the joint i actuating system. The dynamics of the solenoid system A or B are represented using Newton's law applied to the plunger A or B belonging to the solenoid device, as shown in Figure 4 (Hosseini, Arzanpour, Golnaraghi & Parameswaran, 2013),

$$F_{S_{H_i}} = -\frac{\alpha I_{coil_{H_i}}^2}{(\beta + \gamma x_p)^2} = m_p \ddot{x}_p + \lambda_p \dot{x}_p + K_p x_p \quad (5)$$

where m_p is the mass of the plunger, while λ_p and K_p are the stiffness and the damping of the plunger.

The coefficients α, β and γ are constant and depend on the permeability coefficients, effective magnetic flux areas, magnetic field lengths, and turn numbers. These coefficients are parametrized by selecting two points in the $(F_{S_{H_i}}, x_{p_{H_i}})$ space for the joint i . The variable $I_{coil_{H_i}}$ (again, H represents either A or B) is the

current passing through the solenoid and $x_{p_{H_i}}$ is the displacement or stroke of the plunger. Finally, $F_{S_{H_i}}$ is the solenoid force generated by the plunger.

The hydraulic equation links spool displacement x_i and the pressure and flow rate inputs/outputs of the valve i.e. (Q_A, P_A) , (Q_B, P_B) , (Q_P, P_P) and (Q_T, P_T) for joint i . Here, x_i depends on the displacement of plunger A and B, i.e. $x_{p_{A_i}}$ and $x_{p_{B_i}}$, is defined as follows:

$$x_i = x_{p_{A_i}} - x_{p_{B_i}} \quad (6)$$

The volume flow between valve ports in the spool valve is calculated using Bernoulli's law. In particular, for a positive deflection x_i , flows from ports P to A and B to T are as follows (Watton, 2014):

$$Q_A = Q_{P \rightarrow A} = \begin{cases} C_q \cdot A(x) \cdot \text{sgn}(P_S - P_A) \cdot \sqrt{\frac{(2|P_S - P_A|)}{\rho}} & \text{if } 0 < x < w \\ C_q \cdot A_{max} \cdot \text{sgn}(P_S - P_A) \cdot \sqrt{\frac{(2|P_S - P_A|)}{\rho}} & \text{if } x \geq w \end{cases} \quad (7)$$

$$Q_B = Q_{B \rightarrow T} = \begin{cases} C_q \cdot A(x) \cdot \text{sgn}(P_B - P_T) \cdot \sqrt{\frac{(2|P_B - P_T|)}{\rho}} & \text{if } 0 < x < w \\ C_q \cdot A_{max} \cdot \text{sgn}(P_B - P_T) \cdot \sqrt{\frac{(2|P_B - P_T|)}{\rho}} & \text{if } x \geq w \end{cases} \quad (8)$$

where P_A, P_B, P_S, P_T are the pressure values at ports A, B, P and T respectively and sgn is the sign function. Here, C_q is the contraction flow coefficient of the valve, w is the width dimension of the slot in the valve's sleeve, $A(x)$ is the restrictor area of the corresponding orifice and A_{max} is its maximum value. The parameter $A(x)$ is linked to the valve geometry architecture, so it is not possible to derive $A(x)$ as a function of x when no information is available about it. Here, it is initially assumed that there is no leakage flow, no viscous forces, negligible gravity and fluid is flowing at low speed. However, these parameters can optionally be defined in the corresponding modelling elements for the hydraulic actuator.

The final elements of the hydraulic model are the linear and rotary cylinders. Joints 1, 2, 3, 5 and 7 on each manipulator are actuated with a linear double acting cylinder. Assuming zero mechanical friction and gravity effects, the force balance on axis x_i of the cylinder rod is:

$$F_R = A_A P_A - A_B P_B - F_c = (M_R + M_L) \ddot{x}_i + B_D \dot{x}_i + K_S x_i \quad (9)$$

$$F_c = \begin{cases} (x_i - S) \cdot K_p \cdot v & x \geq S, v > 0 \\ x_i \cdot K_p \cdot v & x \leq 0, v < 0 \end{cases}$$

where F_R is the rod force, F_c is hard stop force, K_p is penetration coefficient, v is Cylinder rod velocity, P_A and P_B are the outlet pressure of the valve system, A_A and A_B are the rod areas in chambers A and B respectively, M_R is the mass of the rod, M_L is the mass of the load system, x_i is the rod displacement at joint i , K_S is the load spring constant and B_D represents the viscous damping. For the present research, these hydraulic actuator equations are implemented using the MATLAB SimHydraulics library.

2.3. Mechanical Model Subsystem

The mechanical subsystem involves three elements, as illustrated in Figure 2. The input of the mechanical subsystem is the actuating torque applied to each joint and its output is the angular position of each joint. The kinematic equations of the manipulator are derived using the coordinate systems and simplified serial link conventions presented in Figure 5. Using Figure 6 (a), for each link i except the final one, the force and moment balance equations around the centre of gravity G_i are:

$$\vec{F}_{J_i/0} - \vec{F}_{J_{i+1}/0} + m_{link_i} \vec{g} - m_{link_i} \frac{d}{dt} (\vec{V}_{G_i/0}) = \vec{0} \quad (10)$$

$$\sum \vec{M}_{joints \rightarrow link_i \langle 0 | G_i \rangle} - \frac{d}{dt} (\vec{I}_{link_i \langle G_i | (\bar{x}_i, \bar{y}_i, \bar{z}_i) \rangle} \cdot \vec{\Omega}_{i/0}) = \vec{0} \quad (11)$$

$$\text{with } \sum \vec{M}_{joints \rightarrow link_i \langle 0 | G_i \rangle} = \vec{M}_{J_i \langle 0 | O_i \rangle} + \vec{G}_i \vec{O}_{i/0} \wedge \vec{F}_{J_i/0} - \vec{M}_{J_{i+1} \langle 0 | O_{i+1} \rangle} + \vec{G}_i \vec{O}_{i+1/0} \wedge (-\vec{F}_{J_{i+1}/0})$$

where $\sum \vec{M}_{joints \rightarrow link_i \langle 0 | G_i \rangle}$ is the sum of momentums at point G_i in terms of the base frame, using the

Moments Transport Theorem (Tenenbaum, 2004) to transfer the momentum form point O_i to point G_i .

Furthermore, in equation (10), $\vec{F}_{J_i/0}$ and $\vec{F}_{J_{i+1}/0}$ are the force vectors, $\vec{M}_{J_i \langle 0 | O_i \rangle}$ and $\vec{M}_{J_{i+1} \langle 0 | O_{i+1} \rangle}$ are the

momentum vectors applied at points O_i and O_{i+1} respectively and represented in terms of the base frame,

m_{link_i} is the mass of the link i , \vec{g} is the gravity field vector, $\vec{V}_{G_i/0}$ is the velocity and acceleration of the

centre of mass, represented in terms of the base frame. In equation (11), $\bar{\mathbf{I}}_{link_i(G_i|(\bar{x}_i, \bar{y}_i, \bar{z}_i))}$ is the inertia tensor for link i in frame $(G_i|(\bar{x}_i, \bar{y}_i, \bar{z}_i))$, represented around an arbitrary rotational axis vector $\vec{\Omega}_{i/0}$.

As indicated in Figure 6 (b), the force and momentum balance equations for the last link are:

$$\vec{\mathbf{F}}_{J_i/0} + m_{link_i} \vec{\mathbf{g}} - m_{link_i} \frac{d}{dt} (\vec{\mathbf{V}}_{G_i/0}) = \vec{\mathbf{0}} \quad (12)$$

$$\sum \vec{\mathbf{M}}_{joints \rightarrow link_i(O_i|G_i)} - \frac{d}{dt} (\bar{\mathbf{I}}_{link_i(G_i|(\bar{x}_i, \bar{y}_i, \bar{z}_i))} \cdot \vec{\Omega}_{i/0}) = \vec{\mathbf{0}} \quad (13)$$

where,

$$\vec{\mathbf{M}}_{J_i(O_i|O_i)} + \vec{G}_i O_{i/0} \wedge \vec{\mathbf{F}}_{J_i/0} - \frac{d}{dt} (\bar{\mathbf{I}}_{link_i(G_i|(\bar{x}_i, \bar{y}_i, \bar{z}_i))} \cdot \vec{\Omega}_{i/0}) = \vec{\mathbf{0}} \quad (14)$$

This leads to a total of 36 equations (6 for each insulated link). Since the analytical solution would be extremely complex, a numerical approach using the MATLAB SimMechanic toolbox is adopted. The left and right manipulator mechanical subsystems have the same serial structure i.e. a succession of revolute joint blocks and link subsystems (Figure 7). Revolute joint blocks are actuated by torques supplied by the hydraulic actuator and the angle is sensed as the output variable. Figure 8 shows the geometrical information and properties of link i imported from the available CAD model, transformation and rotation blocks to build a reference frame for link i , and the base frame for the revolute joints i and $i + 1$.

To find the base parameters suitable for estimation, it is assumed that the inertia parameters for each link are calculated using the CAD file information. Therefore, it would be reasonable to consider the mass for each link, and friction and stiffness for each revolute joint, as the unknown parameters for subsequent optimisation. Finally, a 3D representation of the manipulator is shown in Figure 9 using MATLAB Mechanical Explorer. Incorporating the CAD model of the manipulator in this way allows for straightforward understanding of the manipulator behaviour. For example, problems such as collision detection and dual-manipulator coordination can be examined using this visualisation.

3. TWO-STAGE PARAMETER ESTIMATION

Due to the age of the manipulators, data sheets containing values for many of the parameters alluded to above are unavailable, and certain parameters quoted by the manufacturer have undoubtedly changed over time through wear and tear, in addition to replacement parts and other practical matters arising. In part for these reasons, in this section we formulate the estimation problem to be solved using GAs.

Putting all the parameters of the mechanical and hydraulic subsystems together would lead to a very large search space, making such a high-dimensional and nonlinear estimation problem a challenging task.

Instead, by adopting a sequential identification procedure, the estimation is performed in two steps.

The first step is to estimate the parameters of the mechanical subsystem, assuming that the hydraulic subsystem is represented by a scalar gain (K). This ensures that the unknown parameters of the hydraulic model do not affect the tuning of the mechanical system. To illustrate the approach, the focus here is on Joint 2 of the right hand side manipulator (Figure 1). To determine the base parameters of the mechanical subsystem, a sensitivity analysis is conducted (see section 3.1 for discussion of the results). For this purpose, the spring stiffness (S) and damping coefficient (D), as well as the mass of the six manipulator links ($M_i, i=1, \dots, 6$, with link 6 connecting to the gripper) are assumed as the set of parameters for estimation. Note that the exact masses of the individual links are unavailable but their geometry is known via the previously developed CAD models. Hence, by adjusting the estimated mass, the inertia of each link can be tuned, which proves to have a significant impact on the performance.

The parameters of the hydraulic subsystem are estimated as the second step. For this purpose, the parameters of the mechanical subsystem are set as fixed values. For the hydraulic subsystem, there are too many parameters to include in the GA optimisation without encountering significant identifiability and computation issues. Hence, a preliminary simulation study is first completed to identify the key parameters. These results show that the ten most significant parameters to be considered by the GA are:

- The piston area A_A in equation (9) opposite the rod.
- The piston area A_B in equation (9) at the opposite side of the hydraulic actuator, namely the side with the rod hence with a smaller area.
- The maximum ($I_{max_{H_i}}$) and minimum ($I_{min_{H_i}}$) current supplied to the hydraulic valve proportional amplifier in equation (4). These are used to adjust the gain provided by the amplifier, directly affecting solenoid movement which subsequently controls oil flow into the actuators.
- Two constants (A_i and C_i) in equations (2) and (3) used to scale the voltage input between physical limits. On the real device these are used to remove dead-zones from the input. For example, below a certain voltage there is no joint movement, and above a certain voltage the system becomes saturated, hence the input is scaled between these points.
- The solenoid force ($F_{S_{A_i}}, F_{S_{B_i}}$) and stroke ($x_{p_{A_i}}, x_{p_{B_i}}$) in equation (5) are both parameterised with two coefficients each. These parameters are used to generate a force stroke curve for the solenoid, so that given an input current a solenoid displacement is calculated. The movement of the solenoid controls oil flow to the actuator valves.

3.1. Formulation of the Problem as a Single Objective Optimization

To find a mathematical framework for the estimation problem, it is assumed that the parameters of the mechanical or hydraulic subsystems are considered together as a single vector θ . The input/output data used to solve this estimation problem are the voltage applied to the joint i , i.e. $u_i(k)$, and the joint angle measured from the potentiometer of joint i , i.e. $y_i(k)$. Both experimental and simulation data are sampled at 0.01s intervals. To find suitable parameters that relate $y_i(k)$ and $u_i(k)$, the search space Ω is defined to include all prior knowledge about the physical properties of these parameters. As a result, the search space of parameters for the mechanical subsystem $\Omega_M \subset \mathbb{R}^9$ is defined as:

$$\Omega_M = \{\boldsymbol{\theta} \mid \boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_9), \theta_{min_i}^M \leq \theta_i \leq \theta_{max_i}^M, i = 1, 2, \dots, 9\} \quad (15)$$

and for the hydraulic subsystem it will be $\Omega_H \subset \mathbb{R}^{10}$:

$$\Omega_H = \{\boldsymbol{\theta} \mid \boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_{10}), \theta_{min_i}^H \leq \theta_i \leq \theta_{max_i}^H, i = 1, 2, \dots, 10\} \quad (16)$$

Having specified the parametrization of the mechanical (15) and hydraulic (16) subsystems, the identification error for joint i is defined as:

$$\mathcal{E}_i(k; \boldsymbol{\theta}) = y_i(k) - \hat{y}_i(k; \boldsymbol{\theta}) \quad (17)$$

where $\hat{y}_i(k; \boldsymbol{\theta})$ represents the output of joint i in the model for a specific parameter vector $\boldsymbol{\theta}$. To judge the quality of a particular value of $\boldsymbol{\theta}$, it is necessary to define a suitable error signal measure. Here we choose two types of cost function. The first is defined as the p -norm of the absolute error signal for $p = 1, 2, \infty$:

$$J_N(\boldsymbol{\theta}) = \left(\frac{1}{N} \sum_{k=1}^N |\mathcal{E}_i(k; \boldsymbol{\theta})|^p \right)^{\frac{1}{p}} \quad (18)$$

and the second performance measure is defined as the p -norm of the relative error signal with respect to the p -norm of the measured data:

$$\bar{J}_N(\boldsymbol{\theta}) = \left(\frac{1}{N} \sum_{k=1}^N \left| \frac{\mathcal{E}_i(k; \boldsymbol{\theta})}{\|y_i\|_p} \right|^p \right)^{\frac{1}{p}} \quad p = 1, 2, \infty \quad (19)$$

Optimality may, therefore, be expressed by selecting the parameter values that yield the minimal value of the performance measure. Hence, in relation to equation (18), the estimated parameter vector $\hat{\boldsymbol{\theta}}_N$ is:

$$\hat{\boldsymbol{\theta}}_N = \arg \min_{\boldsymbol{\theta} \in \Omega_M} \frac{1}{N} \sum_{k=1}^N |\mathcal{E}_i(k; \boldsymbol{\theta})|^p \quad p = 1, 2, \infty \quad (20)$$

A similar argument applies to the cost function (19).

Before developing a numerical algorithm to solve the optimization problem, an identifiability analysis on the influence of the parameters selected for optimization on the model output $\hat{y}_i(k; \boldsymbol{\theta})$ is carried out. For this purpose, and since the focus here is on joint 2, the sensitivity of $\hat{y}_2(k; \boldsymbol{\theta})$ for different values of the parameters in $\boldsymbol{\theta}$ is evaluated and plotted numerically. To illustrate, Figure 10 shows the sensitivity of selected parameters for a segment of one experiment. For joint 2, the results show that the

output $\hat{y}_2(k; \boldsymbol{\theta})$ is highly sensitive to the damping and stiffness and hence these are identifiable. By contrast, the sensitivity with respect to the masses is not as high and, in fact, reduces by moving from the mass of link 6 to the mass of link 1. This problem is addressed by noting that the sensitivity of $\hat{y}_2(k; \boldsymbol{\theta})$ with respect to the parameters in the vector $\boldsymbol{\theta}$ is a function of time. This means it is possible to find time segments in which the output $\hat{y}_2(k; \boldsymbol{\theta})$ is more sensitive to some specific parameters than other parameters. In fact, this property is used in the multi-objectivization of the output error performance index to improve the parameter estimation accuracy when using the proposed multi-objective GA algorithm. This will be explained in more detail below.

Finally, the non-convexity of the problem is investigated by plotting the performance surface $\bar{J}_N(\boldsymbol{\theta})$ for the two and infinity norms as functions of the parameters. For example, Figure 11 shows the search landscape for an illustrative pair of hydraulic subsystem parameters. Figure 11 and similar plots for other parameters show that the search landscape is not smooth, a situation that is worse when considering the combined dynamics of the actuator and mechanical subsystems.

3.2. Formulation of the Problem as a Multi-Objective Optimization

In section 3.1, the problem of finding an estimate $\hat{\boldsymbol{\theta}}$ of the parameter vector $\boldsymbol{\theta}$ was formulated as minimization of the objective function $J_N(\boldsymbol{\theta})$ in equation (18) or $\bar{J}_N(\boldsymbol{\theta})$ in equation (19). To improve the estimation accuracy of the parameters, our approach in determining the global optimum solution is to use the so called ‘multi-objectivization’ method (Jensen, 2004; Lochtefeld & Ciarallo, 2015). In this technique, a single-objective problem is first formulated as a multi-objective optimization problem and, by solving the reformulated problem, it is possible to provide a solution which minimizes the original single-objective cost function. It is proved that multi-objectivization outperforms single-objective methods on average (Lochtefeld & Ciarallo, 2015). In the present article, the concept is formulated and evaluated in the context of nonlinear parameter estimation for robotic manipulators. To explain the rationale, consider the following definition.

Definition 1 (Marler & Arora, 2004). Assume $\mathbf{F}(\boldsymbol{\theta}) \in \mathbb{R}^n$ is a multi-objective function with

$\mathbf{F} = [F_1(\boldsymbol{\theta}) \quad F_2(\boldsymbol{\theta}) \cdots F_n(\boldsymbol{\theta})]$. The point $\mathbf{F}^0 \in \mathbb{R}^n$ is called a utopia point if and only if

$$F_i^0 = \min\{F_i(\boldsymbol{\theta}) | \boldsymbol{\theta} \in \Omega_M\} \text{ for } i = 1, 2, 3 \cdots n.$$

Here, by splitting the measured output into segments, the single objective optimization problem is converted to the minimization of a multi-objective cost function as follows:

$$J_{\text{mul}}(\boldsymbol{\theta}) = [J_{N_1}^1(\boldsymbol{\theta}), J_{N_2}^2(\boldsymbol{\theta}), \dots, J_{N_n}^n(\boldsymbol{\theta})] \quad (21)$$

where $J_{N_s}^s(\boldsymbol{\theta})$ is the objective function for the segment s defined as:

$$J_{N_s}^s(\boldsymbol{\theta}) = \left(\frac{1}{N_s} \sum_{k=1}^{N_s} |\varepsilon^s(k; \boldsymbol{\theta})|^p \right)^{\frac{1}{p}} \quad p = 1, 2, \infty \quad (22)$$

in which $\varepsilon^s(k; \boldsymbol{\theta}) = y^s(k) - \hat{y}(k; \boldsymbol{\theta})$ and $y^s(k)$ is the measured output at segment s . For the present problem, \mathbf{F}^0 corresponds to the value of the objective function $J_{\text{mul}}(\boldsymbol{\theta})$ at the desired $\boldsymbol{\theta}_0$ which is unattainable in general. However, using evolutionary algorithms, an approximation of \mathbf{F}^0 which is as close as possible to \mathbf{F}^0 will be achieved. Such a solution is called a compromised solution and is Pareto optimal. The challenge is how the word *close* is defined for the compromised solution. In the present work, two criteria are considered: 1- sum of the square error of the objective function at the solution point with respect to the utopia point; and 2- sum of the absolute value of the relative error of the estimated parameters. After the GA algorithm has completed a specific number of iterations, a Pareto set indicating the best parameters estimating the Pareto-optimal solution is achieved. The solution is achieved by finding the minimum of $F_i(\boldsymbol{\theta})$ for all Pareto set solutions and $i = 1, 2, 3 \cdots n$. This leads to a best solution for each segment. Subsequently, the average of these n points is taken as the final solution.

4. PROPOSED GENETIC ALGORITHM

The main challenge in dealing with the optimization problem (19) is that the error signal $\varepsilon(k; \boldsymbol{\theta})$ cannot be formulated analytically as a function of the parameter vector $\boldsymbol{\theta}$. This is due to the fact that a closed form representation of the dynamic equations of the 7-DOF manipulator is not mathematically tractable.

Moreover, the performance index $J_N(\boldsymbol{\theta})$ in equation (15) for $p = 1$ and $p = \infty$ is not differentiable. As a result, finding a mathematical expression showing the gradient of the performance surface with respect to the parameter vector $\boldsymbol{\theta}$ is not practical. Hence, motivated by the issues raised in the introduction to the article, an initial straightforward GA and a more complex, multi-objective GA are described.

4.1. Simple Genetic Algorithm

The GA approach is a very well-known, evolutionary computation global optimisation method, based on the biological principle of natural selection, where the fittest individuals will survive and reproduce. In the GA the individual parameters are encoded as strings of numbers called chromosomes and 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 which are subsequently evaluated using a fitness function. The initial population is randomly selected from the search space Ω_M with a uniform probability distribution. Parent selection is based on the weighted roulette wheel with replacement i.e. the selection probability of each individual is proportional to its fitness. There are various methods for selection and reproduction (Affenzeller *et al.* 2009). In the present study, two crossover approaches are compared i.e. uniform crossover and pointwise crossover. Mutation is the final stage, where single elements may be randomly swapped to create a more diverse population. The process is repeated with a new population, until either a minimum fit is reached or a specific number of iterations are passed. Figure 12 shows a flow chart of the GA approach used here (Montazeri & Poshtan 2003, 2009).

In section 5, each component in Figure 12 is investigated to tailor the algorithm to the specific problem at hand. Two coding schemes are investigated, i.e. integer and multivariable binary string (MVBS) (Montazeri & Poshtan, 2003, 2009). The performance of both coding schemes in terms of finding a better approximation of the optimal solution for the estimation problem is investigated. Using either coding scheme, the chromosomes themselves look the same. For example, equation (23) shows the form of the chromosome for optimisation of the mechanical model,

$$\text{Chromosome} = [K D S M_1 M_2 M_3 M_4 M_5 M_6] \quad (23)$$

where K is the gain representing the actuator subsystem, 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. For each iteration of the algorithm, the simulation model utilises the parameters in that chromosome and the output is compared to experimental data. A fitness value is used to assess the strength of the chromosome, in which the closer the simulation output is to the experimental data, the smaller the fitness value. Selection of the fitness function plays an important role in the convergence behaviour of the proposed GA, hence a number of options are evaluated later in section 5.2, to determine the one which is most suitable for the present application.

4.2. Multi-Objective Genetic Algorithm

Key challenges associated with the initial single-objective optimisation problem are: 1- local optima; 2- keeping the diversity of the population at a reasonable level; and 3- ensuring the algorithm identifies good solutions that can later be assembled by crossover. To address these, a multi-objective GA is implemented using the concept of multi-objectivization (section 3.2). The single objective problem (19) is converted to a multi-objective optimization problem by splitting the measured data into a number of segments and defining a similar objective function for each segment. The developed algorithm utilises the well-known NSGA II i.e. based on non-dominant sorting and Pareto optimal solutions. As noted by Lochtefeld & Ciarallo (2015), the majority of multi-objective evolutionary algorithms, especially those used for multi-objectivization, share a more or less similar algorithmic framework to that of the non-dominated sorting GA and, in engineering applications, no single approach is always superior (Zhou *et al.* 2011). The same argument is true for application of evolutionary algorithms in system identification and filter design (Gotmare *et al.* 2016). Selection of a specific method depends on the type of information provided in the problem, the solution requirement and user preference. Hence, formulating the problem based on

NSGA-II in this article facilitates use of existing results in this area but the applicability of other algorithms such as MODE/D and MOPSO will be investigated in future research. The main purpose of the multi-objective GA is to evaluate and order the offspring in a more effective way, with the aim to improve the performance in regard to the *convergence of the estimated parameters to the true values*. The new algorithm is evaluated in section 5 to determine if the additional complexity is worthwhile.

5. PERFORMANCE OPTIMISATION

Further to the fitness function and the coding scheme explained in section 4, the performance of the GA in finding the ‘best’ estimation result for parameters of the developed model is heavily influenced by several other factors, including the crossover rate, crossover type, mutation rate and population length. Hence, it is necessary to train the simple and multi-objective GA with the most appropriate settings for the present identification problem. For the analysis in this section, the estimated parameters are compared with a set of illustrative parameter values, i.e. numerical values that are physically realistic but are not yet optimised for the real device. These values are listed in the first few rows of Table 1, and are used in the model to generate simulation data for simple and multi-objective GA optimisation (i.e. as a surrogate for experimental data but with known parameter values). For this purpose, the mechanical subsystem is initially considered in isolation using a gain K to represent the hydraulics.

This example allows us to evaluate how close the estimated values of the parameters are to the ‘true’ values already set in the mechanistic model. It also makes clear under which circumstances the algorithm has the capability to converge to these real parameter values, rather than generating a solution based on a different local minimum within the global search space. The latter can happen when the fitness value is relatively low, indicating a close match between the simulated and experimental (here also simulated) output response, but for which the estimated parameters remain far from the true values.

Sections 5.1 to 5.3 below initially focus on the simple GA with MVBS coding, before considering the integer coding scheme and the multi-objective GA.

5.1. Evaluation of Different Crossover Types

Initially using MVBS, pointwise and uniform crossover are investigated. For the pointwise crossover scheme, chromosomes are broken into several segments and each two-parent chromosome swaps segments between the 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. For brevity, full details of the simulation results are omitted but the conclusion is that, using the MVBS coding scheme for this particular application, the pointwise approach consistently yields significantly improved performance compared to uniform crossover.

5.2. Evaluation of Different Fitness Functions

One of the most important issues in mathematical optimization is selection of the fitness function and the system identification problem dealt with in the present article is no exception. Selection of a suitable fitness function enables proper discrimination of the individuals using the proposed GA-based method. Of the six possible fitness functions introduced in section 3.1, two are immediately found to yield rather erratic performance. Hence, only four options are considered in more detail. These are based on equation (18) for $p = 1, 2, \infty$, and equation (19) for $p = \infty$. Figure 13 illustrates the simple GA performance for the MVBS coding scheme using each of these fitness functions. Table 1 shows the relative errors of the optimised parameter values and Table 2 the lowest fitness values and error indices.

These results demonstrate that the convergence of the estimated parameters to their true values requires particular attention in the present context. For this example, Figure 13 shows that norm two (the Euclidian norm) reaches the final fitness value with the least number of iterations searched, while the infinity norm takes the longest. Although norm two has the fastest convergence rate, it is evident from Table 2 that it has a relatively high fitness value, whereas the two infinity norm fitness functions yield very low fitness values. The infinity norm (18) results in both the lowest output identification error and lowest parameter estimation error, and so is the most obvious choice for further evaluation below.

5.3. Evaluation of Different Population Size, Crossover and Mutation Rates

For the present application, population size, crossover and mutation rates are found to have a significant effect on the speed of convergence and final achievable value for the fitness function. Comparing various population sizes, the simulation study suggests that a population size of ~ 70 should be utilised since it generates a diverse population and has the capability to reach a lower value for the performance index. With regard to crossover, a value of 1 means that, at every iteration, the parent chromosomes will create new child chromosomes, hence every iteration will contain different chromosomes to the one before. In Figure 14, for example, crossover values of 0.6 and 0.8 take almost the same time to converge while, despite 0.8 having a higher initial value of fitness, it converges to the lowest value. These results suggest that a crossover value of 0.8 tends to yield the most promising performance. This crossover value represents an empirically derived balance between the requirement for a diversity of population and the need to force the output to determine the fittest individuals. Similar exercises are used to determine the mutation rate and the study shows that the value 0.05 yields the best result in terms of a compromise between the diversity of the population and the pressure on the selection of the highest fitness values. Hence, the best tune for the binary coding scheme is listed in the middle column of Table 5.

5.4. Evaluation of Different Coding Schemes

A similar study (i.e. changing one parameter at a time and comparing the learning curves) is carried out for the simple GA with integer coding. In this case only a uniform crossover is considered. The latter results suggest that the best tune for the parameters of the simple GA with integer coding is population size 70, crossover rate 0.8, and mutation rate 0.5. Finally, the performance of the algorithm for the integer and MVBS coding schemes are compared. With the MVBS scheme, the present analysis uses grey code to prevent sudden erratic jumps in the parameters when crossover occurs, allowing the algorithm to explore the whole search space more effectively. Running the simple GA for both coding schemes and with the attained best tune of the parameters, shows that the MVBS scheme consistently performs better

for this robotic manipulator identification problem. The learning curves of the simple GA with the best tune for both MVBS and integer coding schemes are compared in Figure 15. In all cases, the learning curves are plotted after taking the average over 10 runs.

5.5. Evaluating the Parameters of the Multi-Objective GA

Although the simple GA developed in the previous section forms the reproduction engine for the multi-objective optimization algorithm, the best parameters for this algorithm, nonetheless, require further study. This is a direct application of the “No Free Lunch Theorem” in machine learning. Selected results from the authors’ systematic study into this issue are presented in Tables 3 and 4, while the final column in Table 5 summarises the conclusions. For this purpose, the crossover, mutation, and population size of the multi-objective GA are all varied and the sum of relative parameter estimation error, as well as the sum of square errors, are calculated for each parameter. For brevity, the accuracy of the estimated parameters shown in Table 3 is listed only for selected crossover values. By contrast, the sum of square error and output error index are listed in Table 4 for selected crossover, mutation, and population values. The results in Table 3 show that the crossover values 0.8 and 0.4 both give acceptable results in terms of the estimation accuracy. However, from Table 4 it can be inferred that the crossover values 0.6 and 0.4 yield the best results in terms of the mean square error. This necessitates reaching a compromise for the parameters of the multi-objective GA in terms of the parameter estimation accuracy and reaching the minimum value for the sum of square error. Comparing the index values listed in Tables 3 and 4 suggests that the crossover value of 0.4 is the right choice for further study in the present context. Following a similar analysis, a mutation rate 0.2 and population size 20 is found to yield the best performance for the proposed multi-objective GA in order to achieve better estimation accuracy for the parameters.

Finally, to evaluate how the number of segments in the multi-objectivization process affects the performance of the multi-objective GA, the parameter estimation problem is solved for different numbers of segments. In particular, Tables 4 and 6 (see later section 6.1) compare use of two and eight segments,

and show that both the mean square error of the output and the parameter estimation accuracy are significantly improved by having eight objective functions. In fact, use of eight segments is a pragmatic choice based on a visual examination of the experimental time series, as discussed in section 6.1.

6. PARAMETER ESTIMATION RESULTS

Section 6 extends the analysis to the hydraulic model, applies the approach to experimental data, and compares the simple and multi-objective GA approaches.

6.1. Mechanical Subsystem Estimation

Starting with the mechanical model and using the values in Table 5 as the parameters for the simple and multi-objective GA, Figure 16 compares the dynamic response of the simple GA and multi-objective GA optimised model with the simulated data for Joint 2 (see Figure 1, the shoulder joint). Although the present article focuses on Joint 2 as an example, preliminary analysis of the other manipulator joints yields similar results. For example, GA optimisation for Joint 3 (elbow pitch) yields a dynamic response and parameter estimates that lead to outputs similar to those shown in Figure 16. The voltage input used for this simulation experiment, illustrated in Figure 17, is based on the laboratory experiments considered later in the article and represents a practically realisable signal. The input is scaled to lie in the range $[-100, 100]$ representing the maximum power in a negative direction through to $+100$ representing the maximum power in a positive direction.

As Figure 16 shows, both the simple GA and multi-objective GA estimated simulation model output tracks the original simulation output rather well, albeit with a small over shoot at the peaks. However, Table 6 shows that some of the simple GA parameter estimates are significantly distant from their true (simulated) values, motivating the use of the multi-objective GA approach. To implement the latter, the output is split into eight segments, hence eight objectives to minimise, as illustrated in Figure 18. The use of eight segments was found to work well in practice. It matches the number of peaks and

troughs in the time series (when the joint angle was at the greatest deviation from the initial reference point) and is compatible with the sensitivity analysis reported in section 3.1 (e.g. Figure 10).

In the case of the multi-objective GA, after completion of a specific number of iterations, a Pareto set indicating the best parameters estimating the Pareto-optimal solution is achieved. This helps to restrict our attention to the limited set of choices rather than considering the full range of every parameter. The final value of the estimated parameters $\hat{\theta}$ is achieved by finding an estimation of the Pareto-optimal solution that minimizes each of the eight objective functions, and subsequently determining the average. The parameter vector is utilised to generate a new output response, as illustrated in Figure 19 i.e. the eight thin traces show the best output for each segment, while the thick solid trace shows the average of these. The average optimised response follows the simulation data very well for most of the experiment. Table 6 compares the simple and multi-objective GA optimisation results. For this example, it can be seen that the multi-objective GA with eight segments generally yields more accurate parameter estimates. In fact, the sum of the relative error for the simple and multi-objective GA approaches with two and eight segments are 13.46, 12.75, and 5.46 respectively. This demonstrates significant improvement in the estimation accuracy of the parameters for the multi-objective GA with eight segments compared to the others.

6.2. Hydraulic Subsystem Estimation

The optimisation problem for the hydraulic system is investigated using the same GA settings (Table 5) and voltage input (Figure 17) as for the mechanical system. In this case, however, the hydraulic model is used to generate the model response, with the mechanical model parameters fixed at their optimised values from above. Hence, new simulation data are created to allow the performance of the GAs for the hydraulic system parameters to be analysed. In this case, the simple GA yields an optimised response that actually lies closer to the simulation output response than the multi-objective GA case, with a mean square error of 9.2 compared to 27.1 respectively. However, Table 7 shows that the parameters from the

multi-objective GA are closer to the true values than the simple GA case, yielding a lower sum of relative errors between the true and estimated parameter values, i.e. 3.1 and 3.9 respectively.

6.3. Laboratory Data Parameter Estimation and Model Validation

The simple and multi-objective GAs are applied to experimental data collected from the manipulator, as illustrated in Figure 17 and Figure 20. For direct comparison with the simulation benchmark examples above, the shoulder joint is again chosen to demonstrate the feasibility of the GA approach. More information about the robotic platform is provided by Taylor & Robertson (2013), while the following web page has images of the device: <http://www.lancaster.ac.uk/staff/taylorcj/projects/autonomous>. A semi-automated system for calibrating and initializing the robot for open-loop data collection has previously been developed (Taylor & Robertson, 2013). Here, the robot is first manipulated into a suitable configuration using standard proportional controllers. The operator selects from classical step experiments or pseudo-random signals, with an example of the latter illustrated in Figure 17.

Table 8 shows the mechanical model parameter estimates returned by both the simple GA and multi-objective GA. Figure 20 shows that the optimised simulation output for both algorithms generally follows the dynamic behaviour of the device, although there are some significant underestimates of the joint angle in the case of the multi-objective GA. Note that the focus of earlier sections of the article was to train the multi-objective GA for estimating the true parameter values rather than to minimise such output errors. Nonetheless, an accurate output response is also desirable, of course, and the authors are presently investigating such modelling errors with the aim of improving the response.

In regard to the parameter estimates, the results for the mechanical subsystem can be partially validated by consideration of the known total mass of the manipulator i.e. 45kg. In this regard, the sum of the six estimated link masses in Table 8 is 43.9kg and 79.7kg for the multi-objective and simple GA respectively. This result supports the earlier conclusions of the simulation study, i.e. the multi-objective GA provides closer estimates to the true parameter values than the simple GA. Note that the total link mass was not included as a constraint in the optimisation, since the aim of this example is to investigate GA performance for completely unknown parameters.

Finally, Table 9 shows that the parameters optimised using the laboratory data in Figure 20 also yield a satisfactory output response when they are applied to new experimental data in a simple validation exercise. Here, two different input signals are utilised to generate simulation data (with the same set of parameters as optimised above) and the joint angle responses in each case are compared with the equivalent experimental data collected from the real machine.

7. CONCLUSIONS

The problem of dynamic modelling and parameter estimation for a 7-DOF robot manipulator has been investigated. This manipulator has hydraulic actuators and is suitable for decommissioning and manufacturing applications. A mechanistic model of the system was developed and subsequently implemented using MATLAB software. The parameters of this model are subject to change because of device aging and the characteristics of different operating conditions. To overcome this problem for future simulation and control system design work, the article has exploited and refined a GA 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. In the second step, the proposed GA is utilized to estimate the parameters of the model based on measured experimental data. Simple and multi-objective GAs have been compared, with the latter achieving improved accuracy of the estimated parameters by integrating the validation phase inside the algorithm implicitly. The model is presently being extended to address the other joints of the dual-manipulator system. It will be used for simulation and model-based control design, with current research by the authors focusing on high-level control tasks such as automated remote cutting and welding.

ACKNOWLEDGEMENTS

The authors are grateful for the support of the National Nuclear Laboratory (NNL) and the Nuclear Decommissioning Authority (NDA).

REFERENCES

- Affenzeller, M., Winkler, S., Wagner, S., & Beham, A. (2009). *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. CRC Press.
- Al-Dabbagh, R. D., Kinsheel, A., Mekhilef, S., Baba, M. S., & Shamshirband, S. (2014). System identification and control of robot manipulator based on fuzzy adaptive differential evolution algorithm. *Advances in Engineering Software*, 78, 60-66.
- Bakari, M. J., Zied, K. M., & Seward, D. W. (2007). Development of a multi-arm mobile robot for nuclear decommissioning tasks. *International Journal of Advanced Robotic Systems*, 4, 502-524.
- Bingül, Z., & Karahan, O., (2011). Dynamic identification of Staubli RX-60 robot using PSO and LS methods. *Expert Systems with Applications*, 38(4), 4136-4149.
- Bogue, R. (2011). Robots in the nuclear industry: a review of technologies and applications. *Industrial Robot: An International Journal*, 38(2), 113-118.
- Calafiore, G., & Indri, M., (2000). Robust Calibration and Control of Robotic Manipulators. *Proceeding American Control Conference*, Chicago, 2003-2007.
- Coello, C. A. (2006). Evolutionary Multi-objective Optimization: A Historical View of the Field. *IEEE Computational Intelligence Magazine*, February.
- Das, S., & Suganthan, P. N. (2011). Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation*, 15 (1), 4-31.
- Gautier, M. & Poignet, P. H. (2001). Extended kalman filtering and weighted least squares dynamic identification of robot. *Control Engineering Practice*, 9(12), 1361-1372.
- Gautier, M., Janot, A. & Vandanjon, P.-O. (2013). A New Closed-Loop Output Error Method for Parameter Identification of Robot Dynamics. *IEEE Transactions on Control Systems Technology*, 21(2), 428-444.
- Gautier, M., Jubien, A. & Janot, A. (2012). New Closed-Loop Output Error method for Robot Joint Stiffness Identification with Motor Force/Torque data. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 592-597.
- Gotmare, A., Bhattacharjee, S. S., Patidar, R., & George, N. V. (2016). Swarm and Evolutionary Computing Algorithms for System Identification and Filter Design: A Comprehensive Review, *Swarm and Evolutionary Computation*, in press, onlinedoi: 10.1016/j.swevo.2016.06.007.
- Hosseini, A. M., Arzanpour, S., Golnaraghi, F., & Parameswaran, A. M. (2013). Solenoid actuator design and modeling with application in engine vibration isolators. *Journal of Vibration and Control*, 19(7), 1015-1023.
- Jafari, A., Safavi, M., & Fadaei, A. (2007). A Genetic Algorithm to Optimum Dynamic Performance of Industrial Robots in the Conceptual Design Phase. *IEEE 10th International Conference on Rehabilitation Robotics*, Noordwijk, 1129-1135.
- Janot, A., Vandanjon, P.-O., & Gautier, M., (2014). A Generic Instrumental Variable Approach for Industrial Robot Identification. *IEEE Transactions on Control Systems Technology*, 22(1), 132-145.

- Jensen, M. T. (2004). Helper-Objectives: Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimisation. *Journal of Mathematical Modelling and Algorithms*, 3(4), 323–347.
- Khalil, W., & Dombre, E. (2002). Modeling Identification and Control of Robots. New York: Taylor & Francis.
- Krusiński, J., & Jenkins, W. K. (2005). Design and Performance of Adaptive Systems Based on Structured Stochastic Optimization Strategies. *IEEE Circuit and Systems Magazine*, 8-20.
- Kwok, D. P., & Sheng, F. (1994). Genetic algorithm and simulated annealing for optimal robot arm PID control. *IEEE World Congress on Computational Intelligence, Proceedings of the First IEEE Conference on Evolutionary Computation*, Orlando, 707-713.
- Lochtefeld, D. F., & Ciarallo, F. W. (2015). Multi-Objectivization Via Decomposition: An analysis of helper-objectives and complete decomposition. *European Journal of Operational Research*, 243(2), 395–404.
- Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26, 369–395.
- Mavroidis, C., Flanz, J., Dubowsky S., Drouet, P., Goitein, M. (1998). High performance medical robot requirements and accuracy analysis, *Robotics and Computer-Integrated Manufacturing*, 14, 329-338.
- Montazeri, A., & Poshtan, J. (2003). Optimal placement of loudspeakers and microphones in an enclosure using genetic algorithm. *Proceedings of the IEEE Conference on Control Applications*, Munich, 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*, 70(7), 945-953.
- Montazeri, A. & Udo, E. J. (2016). Development of Dynamic Model of a 7DOF Hydraulically Actuated Tele-Operated Robot for Decommissioning Applications. *Proceedings of the American Control Conference*, Boston, MA, July.
- Montazeri, A., Poshtan, J., & Yousefi-Koma, A. (2008). The use of Particle Swarm to Optimize the Control System in a PZT Laminated Plate. *Smart Materials and Structures*, 17(4), 7pp.
- Nyarko, E. K., & Scitovski, R. (2004). Solving the Parameter Identification Problem of Mathematical Models using Genetic Algorithms. *Applied Mathematics and Computation*, 153(3), 651–658.
- Parsopoulos, K. E. & Vrahatis, M. N. (2010). *Particle Swarm Optimization and Intelligence: Advances and Applications*. Information Science Reference.
- Rouvinen, A., & Handroos, H. (1997). Robot positioning of a flexible hydraulic manipulator utilizing genetic algorithm and neural networks. *Proceedings Fourth Annual Conference on Mechatronics and Machine Vision in Practice*, Toowoomba, 182-187.
- Sun, J. & Liu, X. (2013). A novel APSO-aided Maximum Likelihood Identification Method for Hammerstein Systems. *Journal of Nonlinear Dynamic Systems*, 73(2), 449–462.
- Swevers, J., Ganseman, C., De Schutter, J., & Van Brussel, H. (1996). Experimental robot identification using optimised periodic trajectories. *Mechanical Systems and Signal Processing*, 10(5), 561-577.
- Swevers, J., Verdonck, W., & De Schutter, J. (2007). Dynamic model identification for industrial robots—Integrated experiment design and parameter estimation. *IEEE Control System Magazine*, 27(5), 58–71.
- Taylor, C. J., & Robertson, D. (2013). State-dependent control of a hydraulically actuated nuclear decommissioning robot. *Control Engineering Practice*, 21(12), 1716-1725.
- Taylor, C. J., & Seward, D. (2010). Control of a dual–arm robotic manipulator. *Nuclear Engineering International*, 55, August, 24–26.
- Taylor, C. J., Shaban, E. M., Stables, M. A., & Ako, S. (2007). Proportional–Integral–Plus (PIP) control applications of state dependent parameter models. *IMECHE Proceedings Part I*, 221(17), 1019-1032.

- Tenenbaum, R. A. (2004). Vectors and Moments. In *Fundamentals of Applied Dynamics*: Springer, Chapter 2, 28-89.
- Tohme, E., Ouvrard, R., Abche, A., Trigeassou, J.-C., Poinot, T., & Mercère, G. (2007). Methodology to Enhance the Convergence of Output Error Identification Algorithms. *Proceedings of the European Control Conference*, Kos, Greece, July, 5721-5728.
- Yao, L., & Sethares, W.A. (1994). Nonlinear Parameter Estimation via the Genetic Algorithm. *IEEE Transactions on Signal Processing*, 42 (4), 927-935.
- Vicente, H., Ayala, H., & Coelho, L. S. (2012). Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator. *Expert Systems with Applications*, 39(10), 8968-8974.
- Watton, J. (2014). *Fundamentals of Fluid Power Control*. Cambridge University Press.
- Wu, Z., Wang, W., & Bai, Z. (2012). *Improved Genetic Algorithm Optimizing PID Parameters for Electro-hydraulic Servo System*. Vol. 288 of the series Communications in Computer and Information Science, 386-393.
- Ye, M. (2006). *Parameter Identification of Dynamical Systems Based on Improved Particle Swarm Optimization, Intelligent Control and Automation*. Vol. 344 of the series Lecture Notes in Control and Information Sciences, 351-360.
- Young, P.C. (2011). *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner*. Springer-Verlag, Berlin.
- Zhang, G. Q., Li, X., Boca, R., Newkirk, J., Zhang, B., Fuhlbrigge, T. A., & Hunt, N. J. (2014). Use of Industrial Robots in Additive Manufacturing-A Survey and Feasibility Study. *Proceedings of ISR/Robotik 41st International Symposium on Robotics*, Munich, 512-517.
- Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1) 32-49.

Table 1 – Initial versus ‘true’ (simulation) parameters values for the mechanical model, and relative error of the estimated parameters (i.e. difference between the true and estimated values, divided by true value) for selected fitness functions using the simple GA. The parameters are described in section 3.

	Gain	Damping	Stiffness	Mass of link (kg)					
				1	2	3	4	5	6
Parameter	Value								
Initial	0.1	100	20	0.6	1.3	2.8	1.2	2.2	1.7
True	0.2	231.8	53.4	4.6	5.1	22.5	1.6	4.1	4.7
Fitness Function	Relative Error								
Equation (16) with $p = \infty$	0.5	0.26	0.42	0.19	0.05	0.43	0.12	0.22	1.06
Equation (16) with $p = 1$	1.0	0.82	0.43	0.84	0.94	0.96	2.50	1.46	0.72
Equation (16) with $p = 2$	0.5	0.57	0.29	0.23	0.09	0.84	0.93	0.19	0.21
Equation (17) with $p = \infty$	0.5	0.32	0.47	0.23	0.80	0.78	3.93	0.24	0.12

Table 2 – Comparison of each fitness function associated with the simple GA optimisation shown in Table 1. The output error indices are based on the infinity-norm of the difference between the simulated and optimised dynamic response. The parameter error indices are calculated as the sum of absolute relative estimation errors for the set of nine mechanical model parameters.

Fitness Function	Lowest fitness value	Output error index	Parameter error index
Equation (19) with $p = \infty$	-0.1275	0.651	743
Equation (18) with $p = \infty$	-0.6458	0.645	327
Equation (18) with $p = 1$	-836.54	0.85	969
Equation (18) with $p = 2$	-14.5172	0.82	389

Table 3 – Evaluating the effect of crossover on the performance of the multi-objective GA.

	Gain	Damping	Stiffness	Mass of link (kg)						Sum
				1	2	3	4	5	6	
Crossover	Relative Error									Sum
1.0	0.8	0.36	0.27	0.35	0.02	0.28	2.6	0.78	0.48	5.94
0.8	0.05	0.08	0.15	0.22	0.77	0.02	4.05	2.81	0.31	8.46
0.6	0.35	0.26	0.12	0.28	1.03	0.10	4.7	3.05	1.83	11.72
0.4	0.05	0.08	0.15	0.21	0.77	0.03	4.05	2.81	0.31	8.46

Table 4 – Comparison of the mean square error and output error indices for various multi-objective GA settings (selected illustrative results).

Population	Crossover	Mutation	Mean squared error	Output error index
8 segments				
70	0.8	0.05	0.3133	1.322
50	0.8	0.05	11.2987	7.862
20	0.8	0.05	0.2283	1.2141
70	1.0	0.05	0.2217	0.9240
70	0.6	0.05	0.0580	1.2141
70	0.4	0.05	0.0836	1.2157
70	0.8	0.1	0.0693	1.2148
70	0.8	0.2	0.0945	1.2151
2 segments				
70	0.8	0.05	0.1297	1.214

Table 5 – GA settings used to generate the parameter estimation results in section 6 of the article.

Setting	Simple GA Value	Multi-objective GA Value
Coding scheme	Multivariable binary coding	Multivariable binary coding
Crossover rate (Pc)	0.8	0.4
Mutation rate (Pm)	0.05	0.2
Parent selection	proportional	proportional
Crossover type	pointwise	pointwise
Population size	70	20
Fitness function	infinity norm	infinity norm

Table 6 – Comparison of true and estimated parameter values for the simple and multi-objective GA optimisation, for the mechanical model. The relative error is the difference between the true and estimated values, divided by the true value.

	Gain	Damping	Stiffness	Mass of link (kg)						Sum
				1	2	3	4	5	6	
True value	0.2	231.8	53.4	4.6	5.1	22.5	1.6	4.0	4.7	
Simple GA estimates	0.4	343.2	89.4	4.6	10.60	0.80	10.0	19.5	5.9	
Multi-objective 8 segments average	0.1	112.4	37.5	2.3	2.3	13.3	5.1	2.2	4.5	
Relative error										
Simple GA relative error	1	0.48	0.67	0	1.08	0.96	5.25	3.76	0.26	13.46
Multi-objective 2 segments relative error	0.9	0.86	0.48	0.28	0.77	0.01	4.97	2.66	1.82	12.75
Multi-objective 8 segments relative error	0.5	0.52	0.29	0.5	0.55	0.41	2.19	0.46	0.04	5.46

Table 7 – Comparison of true and estimated parameter values for the simple and multi-objective GA optimisation, for the hydraulic model. The parameters in the top row are described in section 3 The relative error is the difference between the true and estimated values, divided by the true value.

Parameter	Piston Area A	Piston area B	$I_{max_{H_i}}$	$I_{min_{H_i}}$	A_i	C_i	$F_{S_{A_i}}$	$F_{S_{B_i}}$	x_{pA_i}	x_{pB_i}	
True value	1000	750	0.9	0.3	1.4	1.2	8	0.7	0.8	12	
Simple GA estimates	1339	596	0.52	0.27	0.90	1.90	4.23	1.89	0.99	6.19	
Multi-objective 8 segments average	1251	1319	1.1	0.3	0.6	1.1	2.6	0.9	0.7	9.9	
Relative error											
Simple GA relative error	0.339	0.205	0.426	0.114	0.359	0.581	0.471	0.698	0.235	0.484	3.9
Multi-objective 8 segments relative error	0.252	0.759	0.222	0	0.571	0.083	0.675	0.286	0.125	0.175	3.1

Table 8 – Estimated parameters returned by the simple and multi-objective GA for the experimental data, for the mechanical model subsystem.

	Gain	Damping	Stiffness	Mass of link (kg)					
				1	2	3	4	5	6
Simple GA estimates	0.1441	90.52	26.34	1.11	7.80	27.73	10.37	21.86	10.82
Multi-objective 8 segments average	0.3581	314.27	39.06	2.97	4.99	16.08	5.78	7.12	6.99

Table 9 – Output error indices associated with the optimised model applied to three different laboratory experiments, i.e. based on different input sequences. The first data set is illustrated by Figure 17 (input) and Figure 20 (output). Only the first data set is used for GA optimisation hence the other two experiments represent test validation experiments. The output error indices are based on the mean absolute error between the simulated and optimised dynamic response.

Data set	Multi-objective	Simple GA
1	1.7528	1.4301
2	2.5661	2.5414
3	1.3659	1.7591

FIGURES

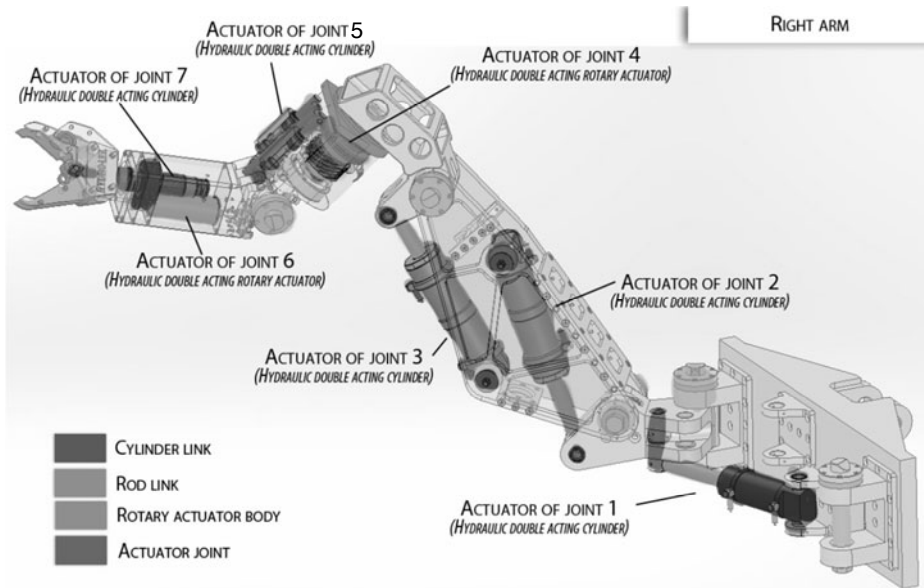


Figure 1 Diagram of 7-DOF (including gripper) manipulator.

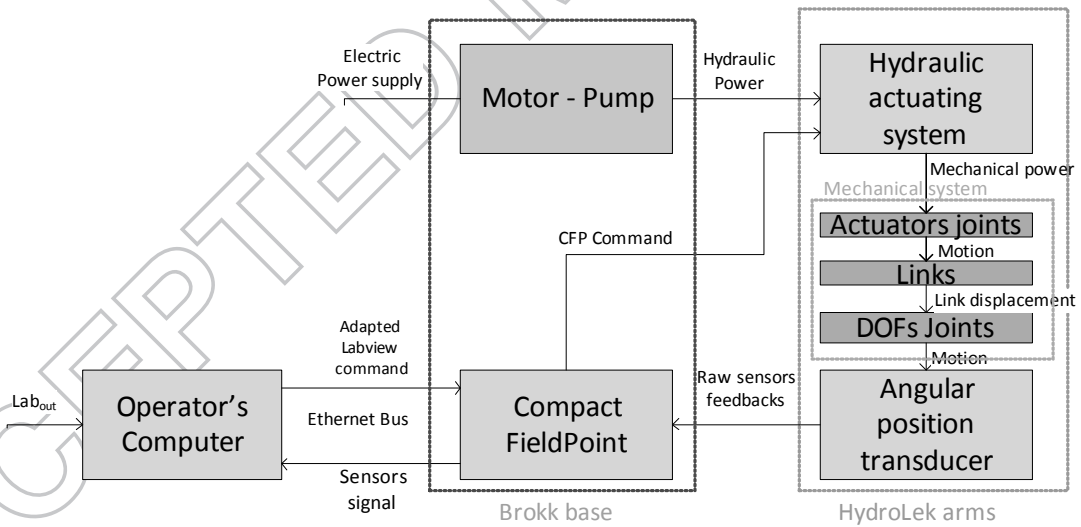


Figure 2 Block diagram of one manipulator adapted for dynamic modelling.

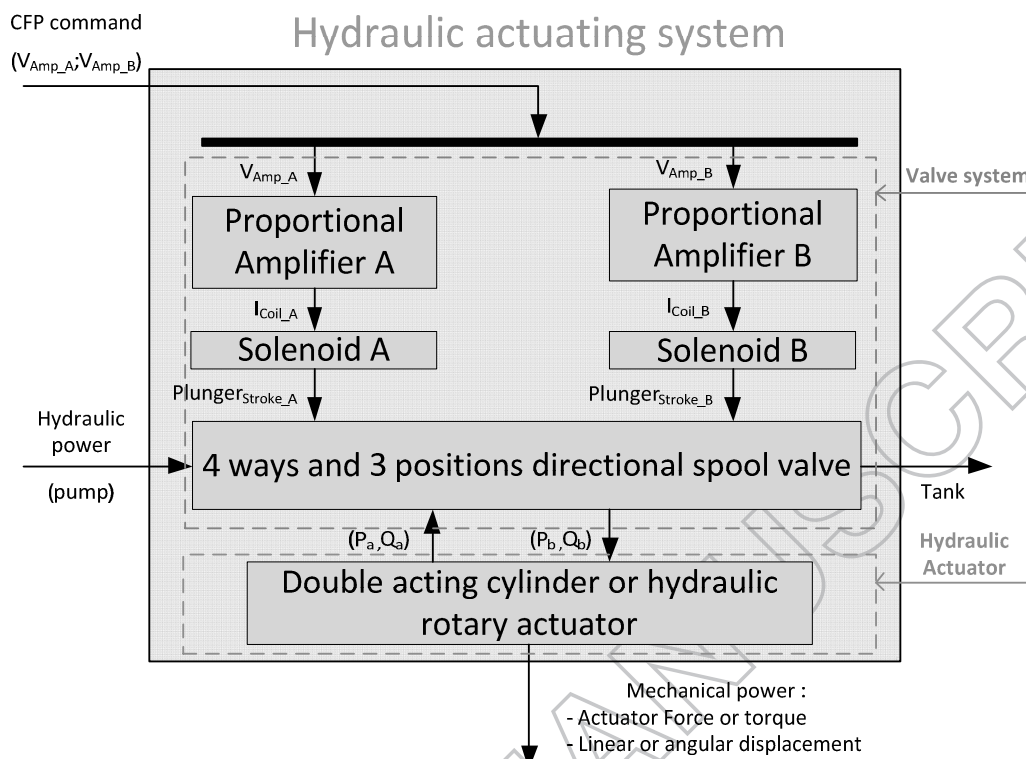


Figure 3 Components of the hydraulic actuating system installed on each joint.

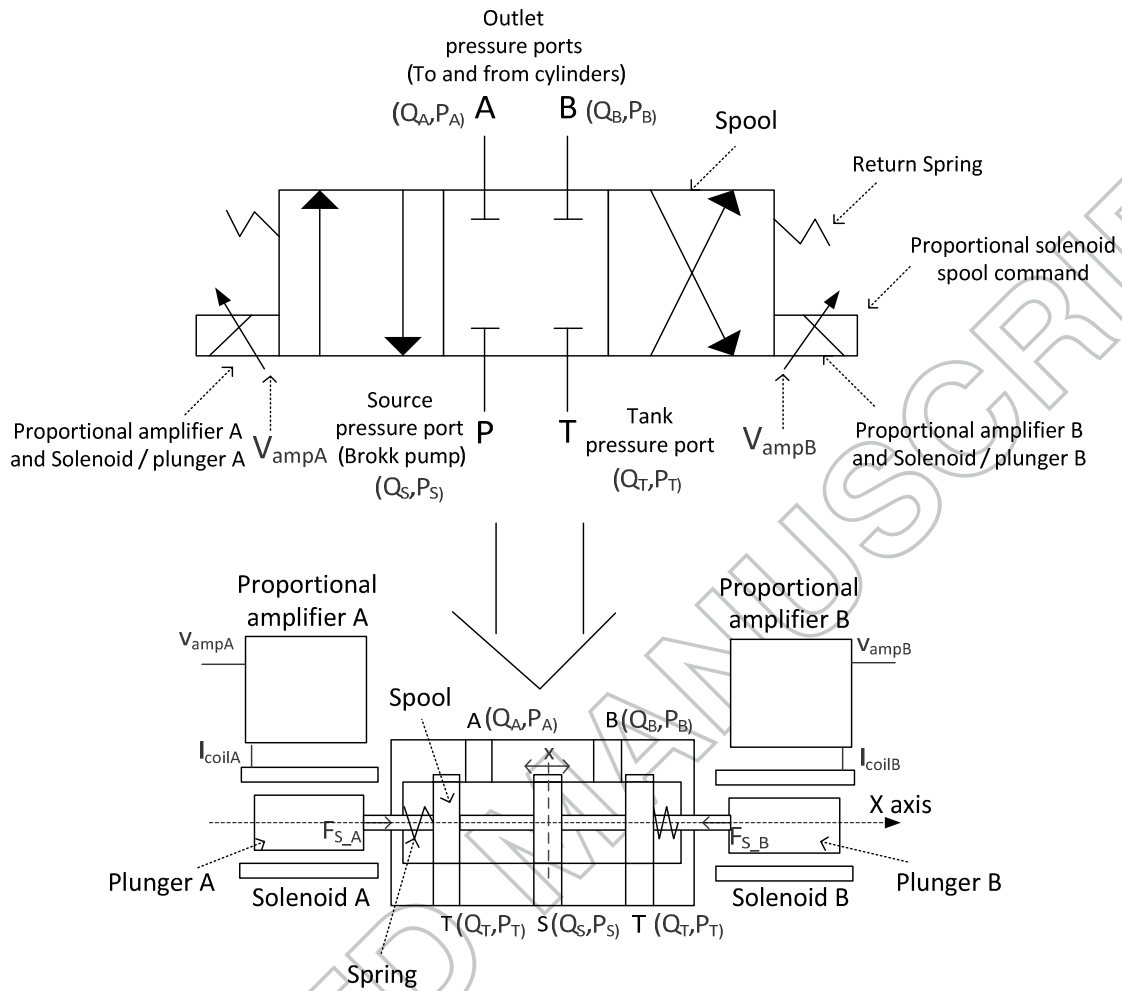


Figure 4 Illustrative model element showing the components of the solenoid device.

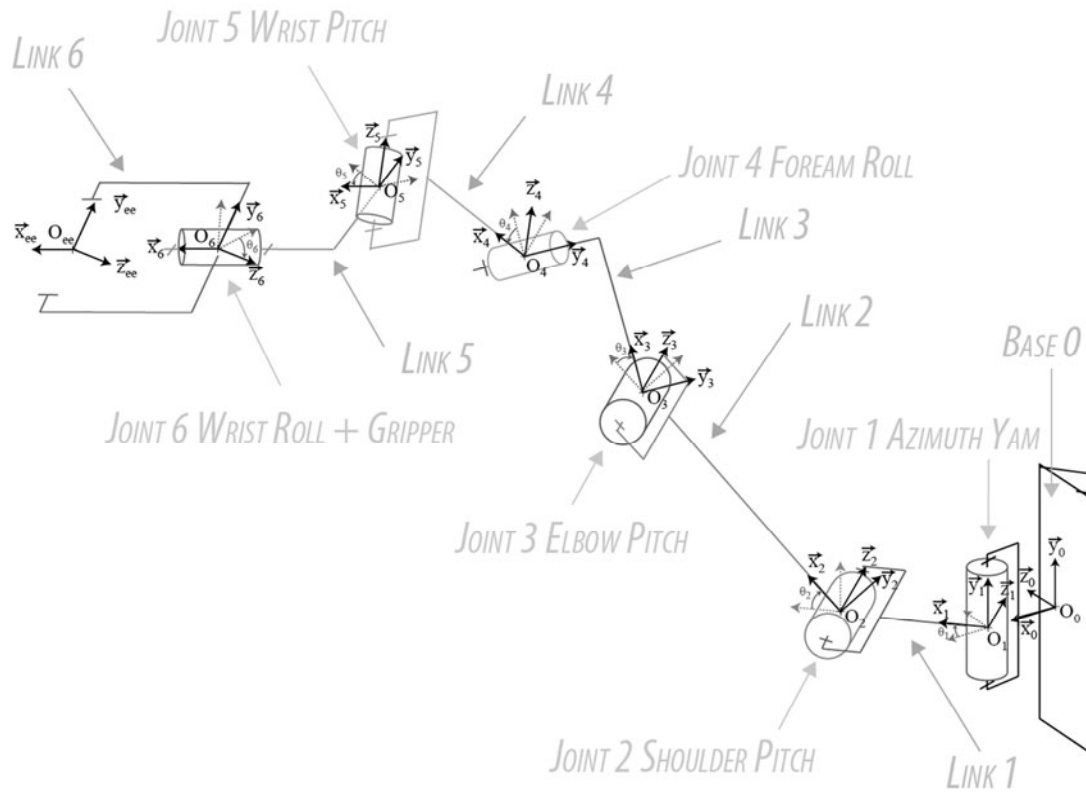


Figure 5 Schematic of the simplified model of HydroLek manipulator.

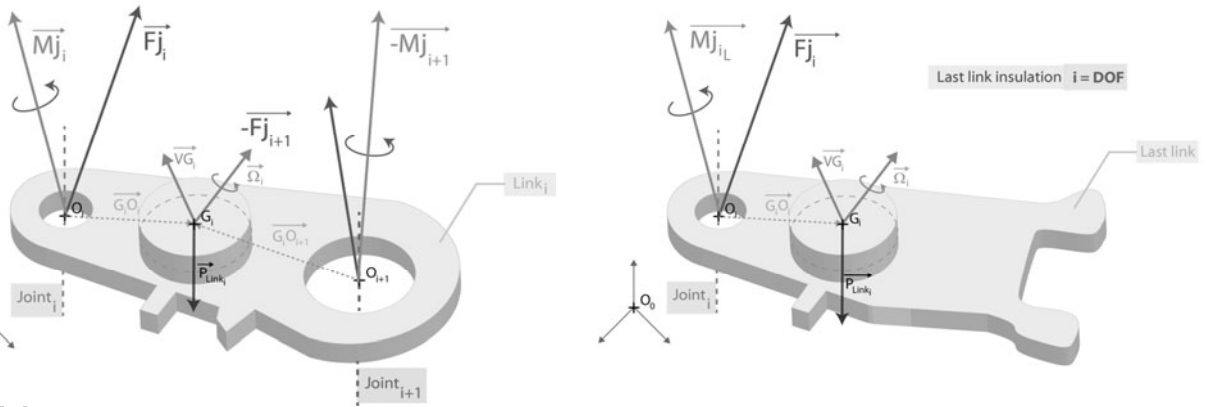


Figure 6 (a) Free diagram body of link i , (b) Free diagram body of the last link.

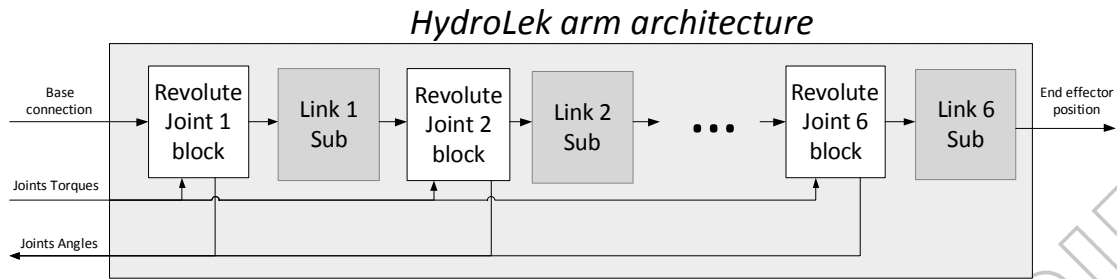


Figure 7 Structure of numerical model of one HydroLek manipulator.

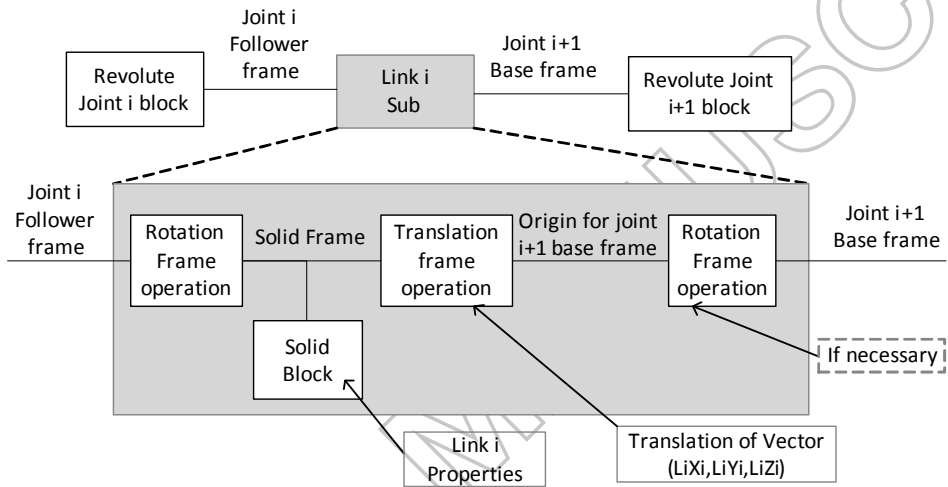


Figure 8 Internal structure of Link i subsystem.

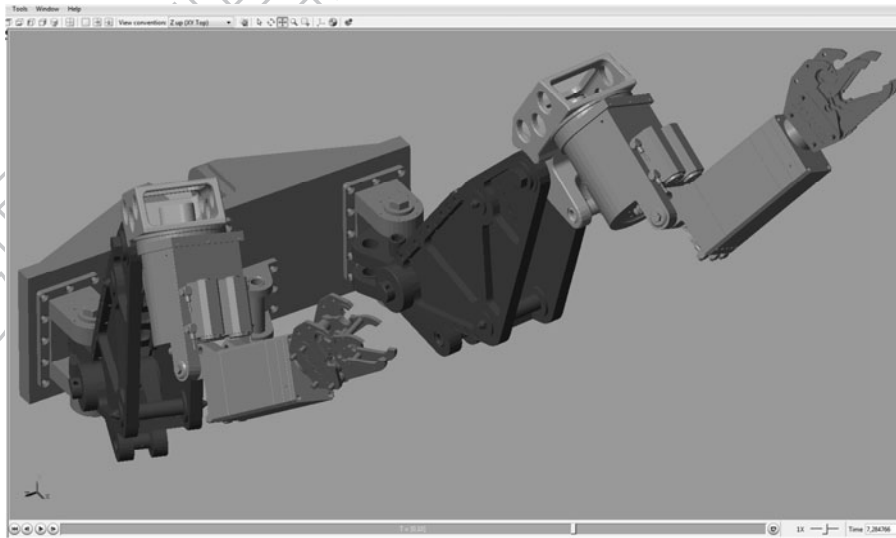


Figure 9 Mechanical Explorer 3D view of the developed manipulator model.

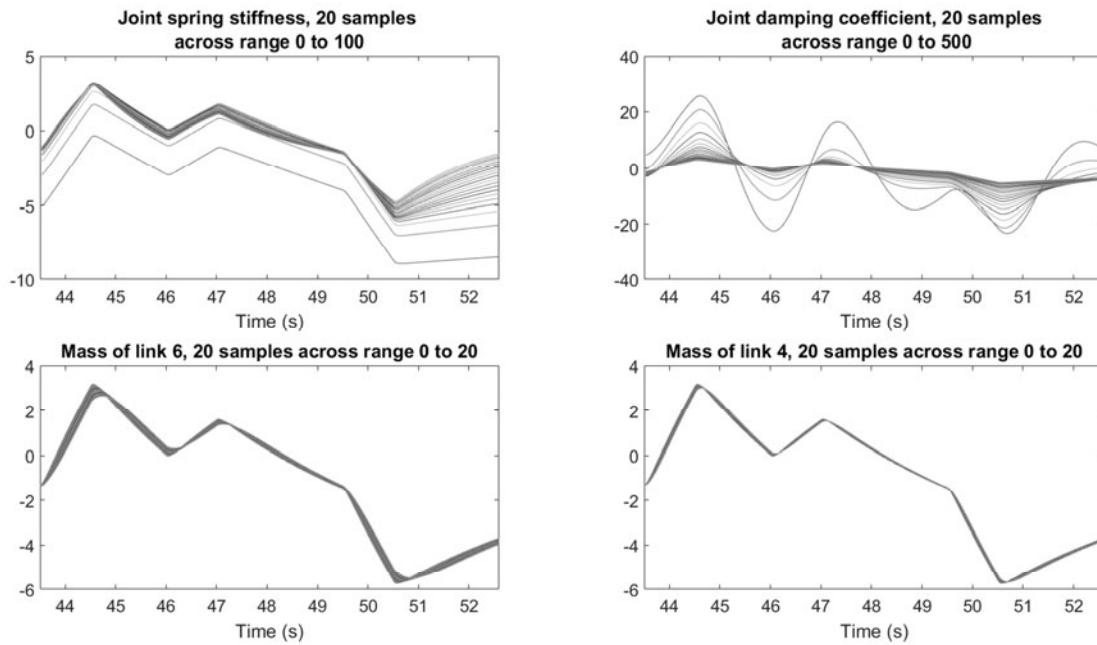


Figure 10 Sensitivity analysis showing the output of Joint 2 around segment 6 (i.e. 43.5s to 52.6s) of an illustrative simulation experiment, for four of the parameters. For each parameter, 20 values within the defined range are plotted.

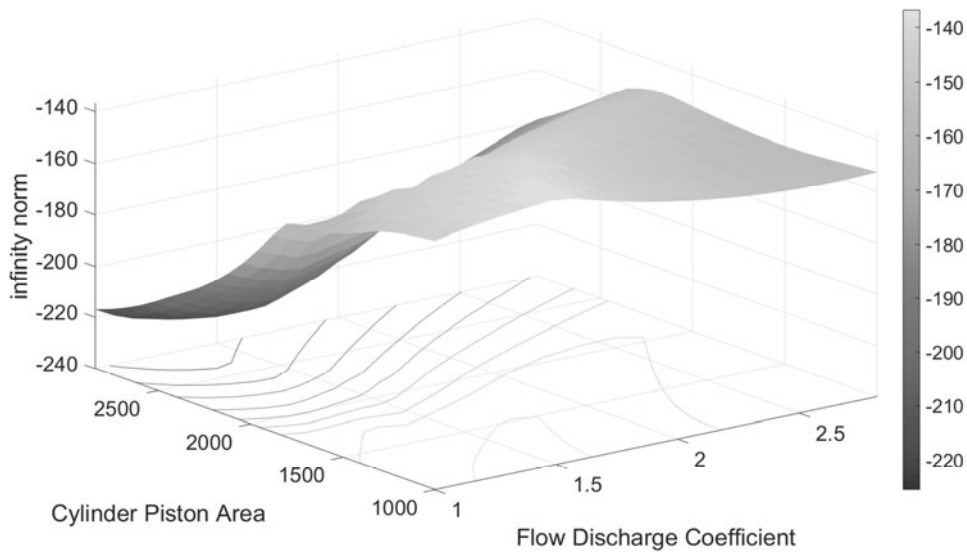


Figure 11 Performance surface showing the infinity norm of the output error as a function of flow discharge coefficient and cylinder piston area of the hydraulic actuator.

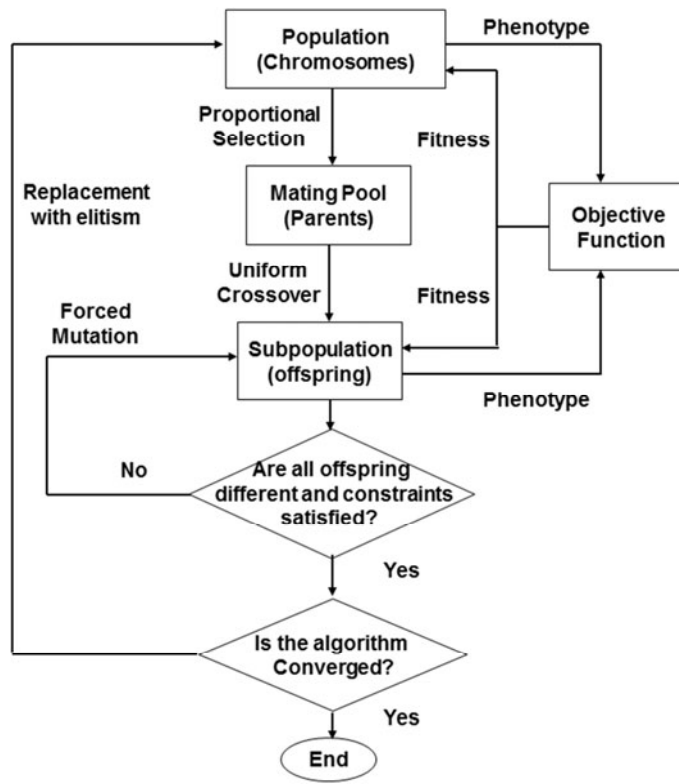


Figure 12 Flowchart showing GA optimisation.

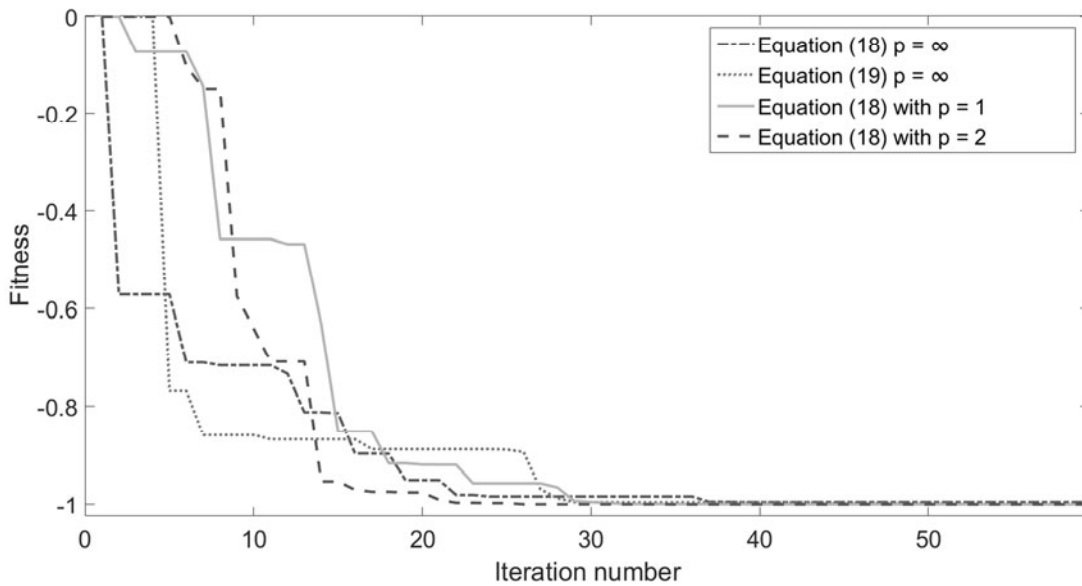


Figure 13 Comparison of four different fitness functions (normalized between 0 and 1) for the simple GA using the MVBS coding scheme.

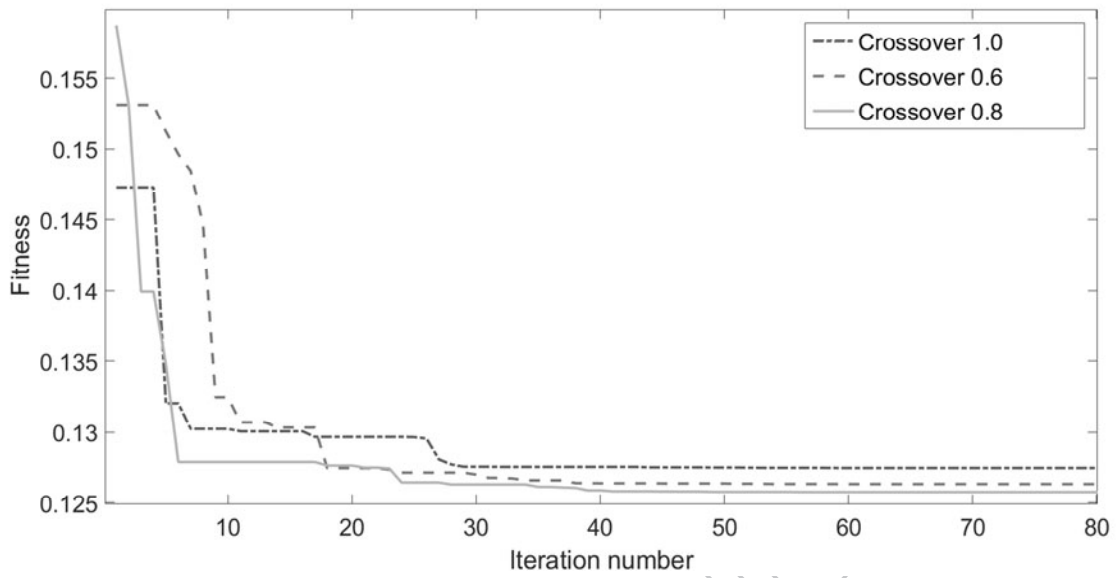


Figure 14 Fitness comparison for three different crossover rates for MVBS coding scheme.

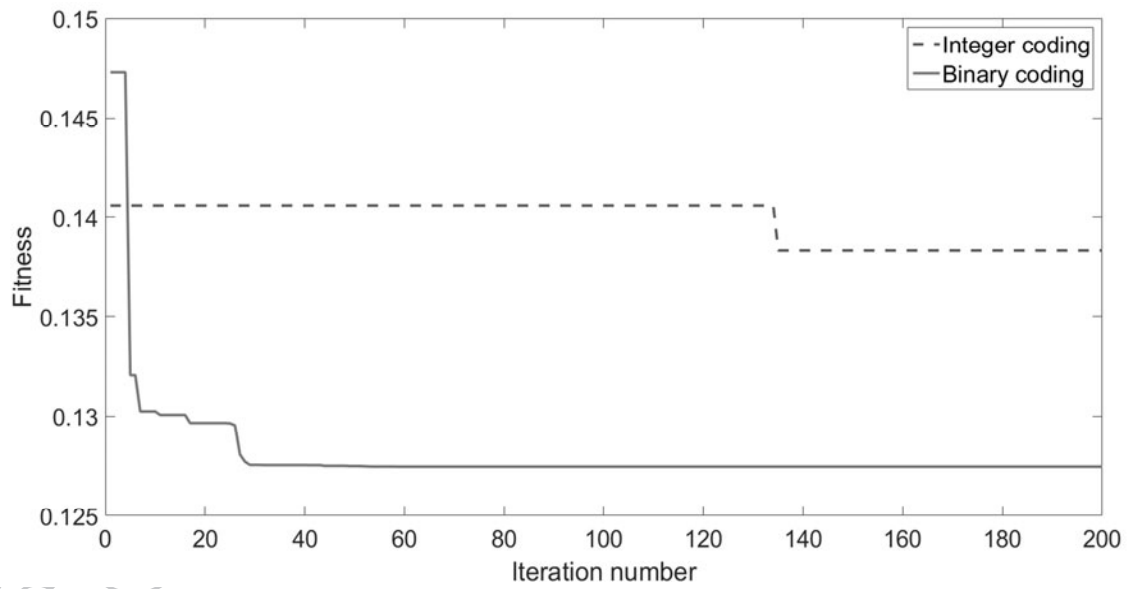


Figure 15 Fitness comparison for binary and integer coding schemes.

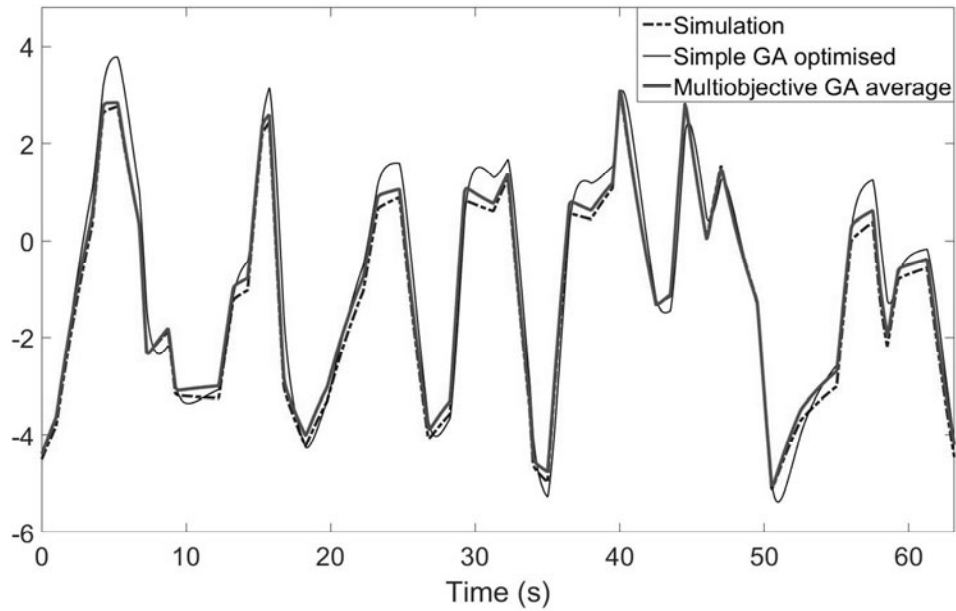


Figure 16 Simulation response of the mechanical subsystem model using the true (dashed trace), simple GA estimated (dot), and multi-objective GA estimated parameters.

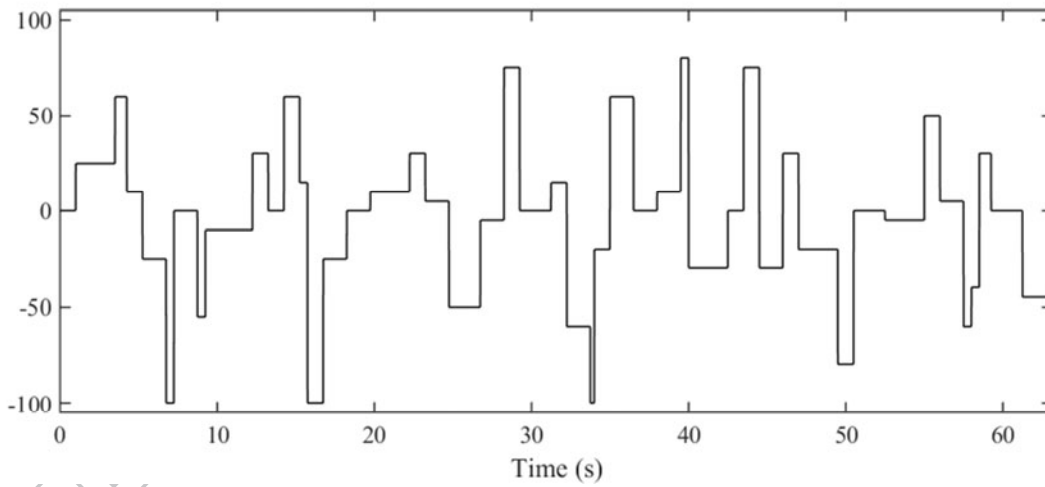


Figure 17 Input voltage used to generate both simulated and experimental data.

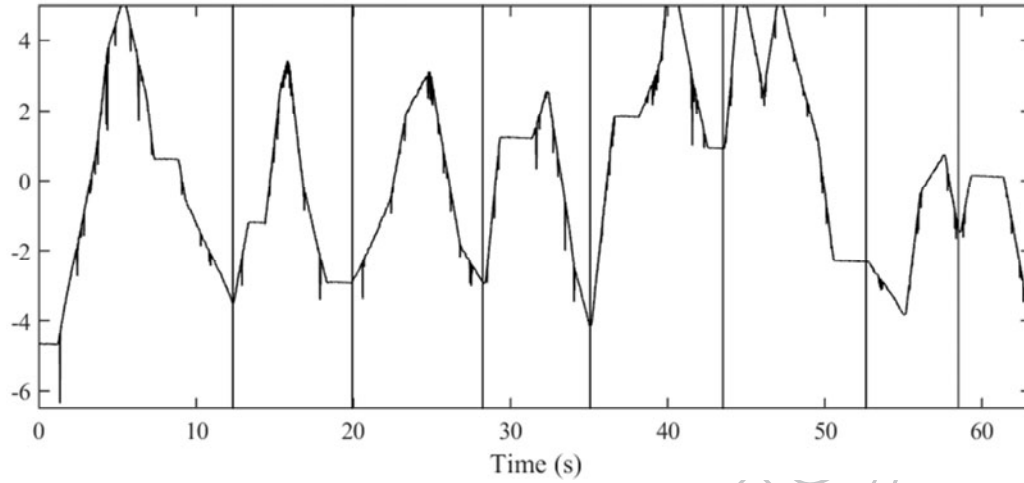


Figure 18 Segmentation of simulation model response.

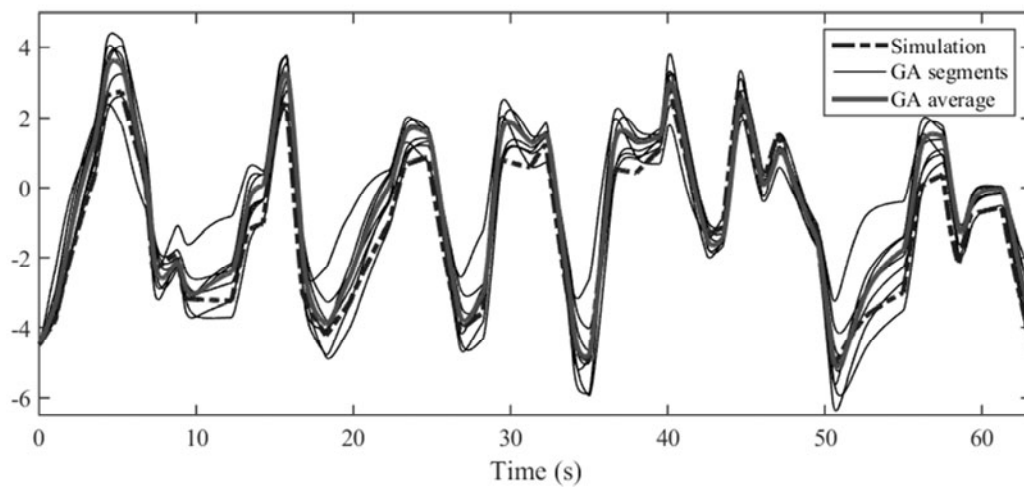


Figure 19 Simulation response of the mechanical subsystem model using true parameters (dashed trace), GA optimisation fits for each of eight segments (thin), and average multi objective GA response (thick).

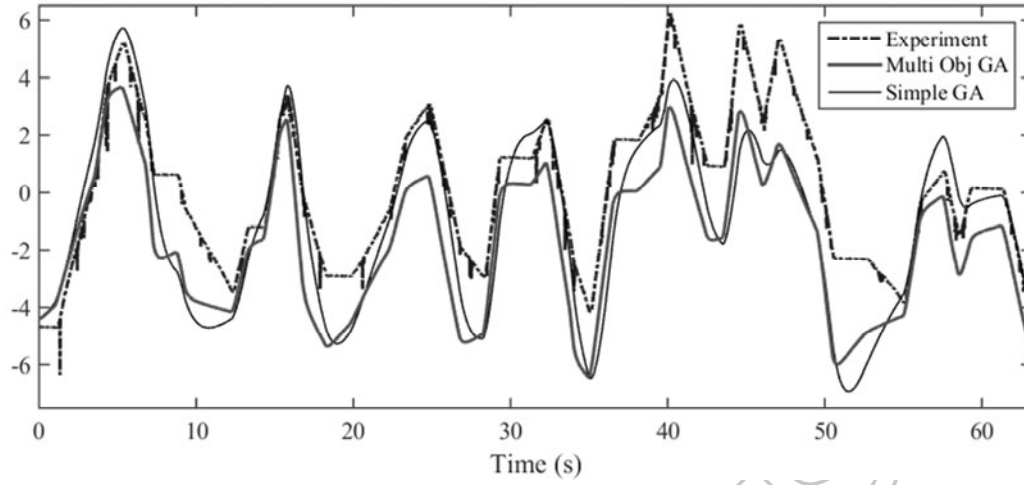


Figure 20 Experimental data from the manipulator (dashed trace), compared with the simulation response for the model optimised using the simple (thin) and multi-objective (thick) GA.

ACCEPTED MANUSCRIPT