IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 25, 2018, accepted October 7, 2018, date of publication October 10, 2018, date of current version October 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2875272

Mechanical Design Optimization of a 6DOF Serial Manipulator Using Genetic Algorithm

EMIL BJØRLYKHAUG[1] AND OLAV EGELAND[2]
[1]Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway
[2]Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway

Corresponding author: Emil Bjørlykhaug (emil.bjorlykhaug@ntnu.no)

This work was supported by the Research Council of Norway under Grant 245613/O30.

ABSTRACT Robots are becoming increasingly more common in the industry. In order to expand the use of robotic manipulators to complex tasks, a higher degree of tailoring of robots may be required. Tailoring of the mechanical design of a robot manipulator can be formulated as an optimization problem and we propose to use Genetic Algorithms (GA) to optimize link lengths, link diameters, and link thickness in order to determine the robot design for a given fitness function. As a case, the optimization of a robot manipulator for automated cleaning of fish processing plants is presented. For this application, the kinematic layout and joint configuration are decided beforehand. The dynamics of the robotic manipulator are presented along with the optimization algorithm and implemented in Java. Results show that the GA is well suited for the optimization of the design.

INDEX TERMS Genetic algorithms, manipulator dynamics, mechanical engineering, optimization.

I. INTRODUCTION

The International Federation of Robotics reports that 72.7% of all industrial robots are used for pick-and-place, welding and assembly [1]. For certain tasks, conventional robot designs as supplied by the major robot companies may not be suited for the task. This motivates the use of custom robot manipulator design, where both the joints and the links are designed with the given task in mind. Robotic systems are inherently complex [2], and tools and techniques that can help simplify the design process are needed.

It is not necessarily straightforward to develop a good design for a given task specification. The problem is multi-objective, multiconstrained and multivariate [3], and the optimization of parameters such as link length, stiffness, total reach and weight are important, and often contradicting. This makes the optimal solution difficult to define, and difficult for a human designer to solve optimally. For instance, optimizing link lengths such that no joints are underpowered compared to the others in a worst case scenario is important for robotic manipulator design, and while optimizing the link lengths, the other link parameters such as topology, diameter and thickness may become unfavourable, resulting in the need to re-optimize these parameters. This will in turn require re-optimization of the link lengths due to changes in weight, stiffness, etc. for the links.

A great deal of interest has been devoted to the optimization of the kinematic chain of robotic manipulators in order to fulfill a desired work envelope, and to achieve good dexterity and high configurability. Kivelä *et al.* [4] optimized the structure of a heavy duty hydraulic serial robotic manipulator to cover required task points. Sun *et al.* [3] optimized the mechanical structure of a grinding robot by reducing the optimum design variables of the search space. Genetic algorithms (GA) have been used to optimize robots. Chocron and Bidaud [5] used a two-level GA to optimize the topology for a modular robot from a given task specification. Chung *et al.* [6] determined the necessary configuration and optimal link length for a modular robot from task specifications. A similar approach was used by Jafari *et al.* [7] where GA was used to find optimal link lengths and optimal gearbox sizing for a 3DOF manipulator when considering the dynamic performance of the manipulator. West *et al.* [8] used GA for estimating parameters for the dynamic modelling of a 7DOF hydraulic manipulator. An evaluation of the influence of encoding scheme, crossover method and crossover rate for the performance of the GA was also performed. Tu *et al.* [9] optimized restoration accuracy of a Stewart platform bumper for strap-down inertial navigation system.

Other soft computing optimization technique and variations of the GA have also been used on robotic manipulators. Wang *et al.* [10] used Multi-Objective Particle Swarm Optimization to optimize the design of a planar parallel 3DOF nanopositioner. Francalanza *et al.* [11] used generative design in the development of a robotic manipulator.

The generative design process was used to produce a mesh for the link topology in order to save weight. Generative design is a process that is similar to GA, in the sense that it is inspired by evolutionary processes. Some implementations of generative design uses GA to create solution proposals. Saravanan *et al.* [12] used three different optimization algorithms, namely; Multi-objective Genetic Algorithm (MOGA), Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) and Multi-objective Differential Evolution (MODE) to optimize the design of robot grippers. GA has also been used for conceptual design. Bentley and Wakefield [13] used GA to develop conceptual designs for optical prisms. A methodology for dimensional synthesis for multiobjective optimization of the linear Delta parallel robot was presented by Kelaiaia *et al.* [14]. They used Strength Pareto Evolutionary Algorithm 2 [15] to find a Pareto front for the different evaluation criteria. In many regards, a crane is similar to a robotic manipulator. Bye *et al.* [16] used GA for optimizing parameters of knuckleboom crane designs in a virtual prototyping tool. However, no existing literature covers the optimization of all the link parameters with respect to payload, reach and end effector stiffness for serial manipulators.

Conventional robotic manipulator designs does not fulfill the requirements of a robotic cleaning system for fish processing plants [17]. Several aspect of the robotic design deserves extra attention for a robotic cleaning system aimed at fish processing plants. Special consideration regarding corrosion resistance, intrinsic contamination, transportation system, etc. has to be taken into account in order to deliver satisfying operating performance. In addition, a robotic manipulator suitable for cleaning of fish processing plants has to have a long reach (>2m), but has lower payload requirement than typical industrial robotic manipulators. The robotic manipulator itself has to have long reach, be slender, have good dexterity, provide adequate payload, while keeping weight as low as possible.

Our novel contribution is to use GA to optimize link length, diameter and thickness for a complete robotic manipulator when the joints and their physical properties are decided beforehand. In addition, the slender design of a robotic manipulator suitable to navigate in narrow spaces and intended for fish processing plant cleaning is presented.

The rest of the paper is organized as follows. First, an example of the robotic manipulator to be optimized is presented, along with insight into the chosen design. This is presented in Section II. The dynamics, mechanical aspect and forces and torques acting on a robotic arm are presented in Section III. In Section IV, the kinematics problem is presented, along with the drawback of using the DH convention when calculating the transformations which should be used when calculating the dynamics of the robot. Section V shows how the parameters of the GA and how the experiment was set up, along the with results from the experiment. Finally, Section VI concludes and discusses further work.

FIGURE 1. Initial design of the robot manipulator.

TABLE 1. Initial DH parameters of the robot.

	d_i [mm]	θ_i [°]	a_i [mm]	α_i [°]
1	-180	0	0	90
2	0	-90	1340*	0
3	0	0	1340*	0
4	0	90	0	-90
5	-290	0	500*	0
6	-67	0	500*	0

II. ROBOT MANIPULATOR DESIGN

The initial design of the robot manipulator to be optimized is shown in Figure 1, and the Denavit-Hartenberg parameters are given in Table 1. The links marked with an asterisk are the ones to be optimized. The purpose of the robot manipulator is automated cleaning of fish processing plants.

A. LINK DESIGN

The links are designed with simplicity of manufacturing in mind. The main part is a cylindrical beam, with end-plates to accommodate the attachment of joints. This experiment is limited to the case where a cylindrical beam is the main link element. This is due to hygienic consideration, as crevices and pockets where water and bacteria can get trapped during operation of the robot, is to be kept to a minimum. More complex topologies such as lattice beam may introduce unwanted risk for enabling bacterial growth. However, the proposed approach for optimizing the design can be extended to include different link topologies.

An exploded view of the link design is shown in Figure 2. For the optimization, the links are assumed to have uniform mass distribution, and a center of gravity in the middle of the link length.

B. JOINTS

The primary operation for the robot arm is to move horizontally in the *xy* plane. For this reason, joints 2, 3 and 4 are primarily for vertical displacement, while 5 and 6 are for the

FIGURE 2. End plate on a link.

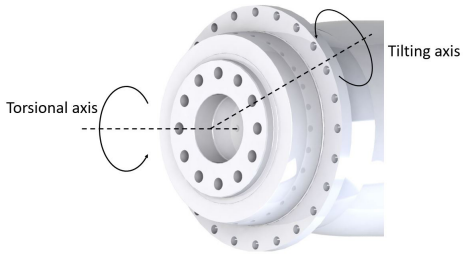

FIGURE 3. The torque axes of a revolute joint.

horizontal displacement. Since the robot has to access small orifices in machines that need to be washed, joint 6 has an actuator that is not mounted on the axis of rotation.

III. DYNAMICS

The mathematics and dynamics of the various parts of a robotic manipulator will be discussed in this section.

A. TORQUE ACTING ON JOINTS

There are two types of torque acting on a joints - A tilting torque acting perpendicular to the rotation axis of the joint, and a torsional torque acting along the rotation axis of the joint, as shown in Figure 3. Both of these torques must be taken into account, as for a given joint, both will have a maximum value. In addition, a given joint will have a stiffness coefficient associated with each of these torques.

B. TORQUE ACTING ON LINKS

Similarly to joints, links can also be affected by torques. Cylindrical joints are a special case where the tilting stiffness is the same for all axes that are perpendicular to the torsional axis. The axes are shown in Figure 4.

C. MAXIMUM FORCE AT END EFFECTOR

One of the most important features of a robotic arm is the payload it can handle. When calculating the maximum payload force at the end effector, the maximum allowable torques on joints and links have to be considered. In addition to the

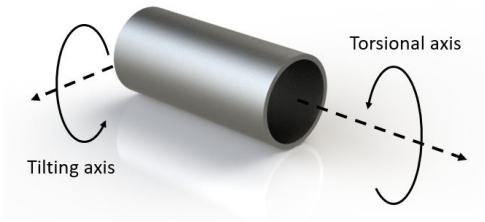

FIGURE 4. Axes of a link.

force acting on the end effector inducing torques on joints and links, the gravitational pull on the masses of the robotic arm itself will affect the payload. Thus, it is possible that one joint is severely limiting the maximum payload compared to the other joints. Optimizing such that that is not the case becomes a complex task as the number of joints and accompanying links increases.

Here we present the calculations for the maximum payload at the end effector. We take a top down approach, beginning with the system seen as a whole. The force

$$F_{max} = \min(F_{max,L_1}, F_{max,J_1}, F_{max,L_2}, F_{max,J_2} \ldots F_{max,L_n}, F_{max,J_n}) \quad (1)$$

is the maximum allowable force on the end effector in the gravitational direction (i.e. payload), limited by the weakest joint or link in the current pose. F_{max,L_i} is the maximum force on the end effector given by link i and F_{max,J_i} is the maximum force on the end effector given by joint i. Here

$$F_{max,L_i} = \min\left(\frac{M_{max,L_i,tilt} - M_{g,L_i,tilt}}{||\mathbf{M}_{z,i,EE} \times \mathbf{z}||}, \frac{M_{max,L_i,tors} - M_{g,L_i,tors}}{||\mathbf{M}_{z,i,EE} \cdot \mathbf{z}||}\right) \quad (2)$$

is the maximum force on the end effector in the gravitational direction limited by link i. Maximum ratings for both torque axes has to be considered. $M_{max,L_i,tilt}$ is the maximum torque link i can handle before buckling. $M_{g,L_i,tilt}$ is the tilting moment induced on link i by the masses of the links and joints which are later in the kinematic chain. $||\mathbf{M}_{z,i,EE} \times \mathbf{z}||$ is the moment arm from link i to the end effector, for forces acting along $\mathbf{z} = [0, 0, 1]^T$, i.e. the gravitational direction. The same has to be calculated for torsional torque, and the smaller value will determine maximum payload when considering link i.

Similarly,

$$F_{max,J_i} = \min\left(\frac{M_{max,J_i,tilt} - M_{g,J_i,tilt}}{||\mathbf{M}_{z,i,EE} \times \mathbf{z}||}, \frac{M_{max,J_i,tors} - M_{g,J_i,tors}}{||\mathbf{M}_{z,i,EE} \cdot \mathbf{z}||}\right) \quad (3)$$

where F_{max,J_i} is the maximum force on the end effector in the gravitational direction limited by joint i. $M_{max,J_i,tilt}$ is the maximum rated torque of joint i. $M_{g,J_i,tilt}$ is the moment induced on link i by the masses of the links and joints which are later in the kinematic chain. $||\mathbf{M}_{z,i,EE} \times \mathbf{z}||$ is the moment arm from joint i to the end effector, for forces acting along $\mathbf{z} = [0, 0, 1]^T$, i.e. the gravitational direction. The same has

to be calculated for torsional torques, and the smaller value will determine the maximum payload for joint i.

The effect that gravity has on a joint in the tilting direction can be calculated as follows:

$$M_{g,J_i,tilt} = \sum_{n=i}^{6} m_{J,n} \times 9.81 \times ||\mathbf{M}_{z,i,n} \times \mathbf{z}|| + \sum_{n=i}^{6} m_{L,n} \times 9.81 \times ||\mathbf{M}_{z,i,n} \times \mathbf{z}|| \quad (4)$$

where $M_{g,J_i,tilt}$ is the sum of all the gravitational moments from joints and link which comes later in the kinematics chain for joint i. m_{J_n} is the mass of joint n, m_{L_n} is the mass of link n, 9.81 is the gravitational acceleration, $\mathbf{z} = [0, 0, 1]^T$ and

$$\mathbf{M}_{z,i,n} = \mathbf{P}_n^i \times \mathbf{z}_i \quad (5)$$

is the moment vector for item i. $\mathbf{P}_n^i$ is a vector from item i, to the center of gravity (COG) for item n. $\mathbf{z}_i$ is in the case of a joint the rotational axis with the current pose. For a link $\mathbf{z}_i$ is the vector from the joint the base of link is attached to, to the next joint, where the end of the link is attached. The equations for calculating the effect gravity has on a joint in the torsional direction, and on links in the tilting and torsional direction, follows:

$$M_{g,J_i,tors} = \sum_{n=i}^{6} m_{J,n} \times 9.81 \times ||\mathbf{M}_{z,i,n} \cdot \mathbf{z}|| + \sum_{n=i}^{6} m_{L,n} \times 9.81 \times ||\mathbf{M}_{z,i,n} \cdot \mathbf{z}|| \quad (6)$$

$$M_{g,L_i,tilt} = \sum_{n=i+1}^{6} m_{J,n} \times 9.81 \times ||\mathbf{M}_{z,i,n} \times \mathbf{z}|| + \sum_{n=i}^{6} m_{L,n} \times 9.81 \times ||\mathbf{M}_{z,i,n} \times \mathbf{z}|| \quad (7)$$

$$M_{g,L_i,tors} = \sum_{n=i+1}^{6} m_{J,n} \times 9.81 \times ||\mathbf{M}_{z,i,n} \cdot \mathbf{z}|| + \sum_{n=i}^{6} m_{L,n} \times 9.81 \times ||\mathbf{M}_{z,i,n} \cdot \mathbf{z}|| \quad (8)$$

D. STIFFNESS AT END EFFECTOR

The calculation of the stiffness for robotic manipulators have received considerable interest in the literature over the years, since it is regarded as one of the most important indicators of performance for robotic systems [18], [19]. In 1980, Salisbury [20] formulated the mapping of stiffness matrices between Cartesian and joint space as:

$$\mathbf{K}_c = \mathbf{J}^{-T}\mathbf{K}_\theta\mathbf{J}^{-1} \quad (9)$$

where $\mathbf{K}_c$ is the Cartesian stiffness matrix, $\mathbf{K}_\theta$ is the joint stiffness matrix and $\mathbf{J}$ is the Jacobian matrix. However, it has been shown that the equation only holds true when the manipulator is at an unloaded equilibrium configuration. When it is not in an unloaded equilibrium configuration the Conservative Congruence Transformation has to be applied to give the correct Cartesian stiffness [21]:

$$\mathbf{K}_c = \mathbf{J}^{-T}(\mathbf{K}_\theta - \mathbf{K}_g)\mathbf{J}^{-1} \quad (10)$$

where

$$\mathbf{K}_g = \left[\left(\frac{\partial \mathbf{J}^T}{\partial \theta_1}\mathbf{f}\right) \left(\frac{\partial \mathbf{J}^T}{\partial \theta_2}\mathbf{f}\right) \cdots \left(\frac{\partial \mathbf{J}^T}{\partial \theta_n}\mathbf{f}\right) \right] \quad (11)$$

defines the changes in geometry in response to an situation where the manipulator is not in an unloaded equilibrium configuration.

However, this approach assumes that the joints only deflect in the torsional axis, and that joints are infinitely stiff in the tilting axis. It also assumes that links are rigid. Since there is no point in having grossly over-dimensioned links, the dynamics of the links also have to be considered, if they are to be optimized. The Virtual joint Modeling method (VJM) is a method of stiffness modelling which can represent the complete stiffness model of a robot, including tilting stiffness of joints and stiffness of links [22]. VJM is a lumped element model where rigid bodies the robot manipulator are paired with virtual joints. An example of a VJM model for a serial manipulator can be seen in Figure 5.

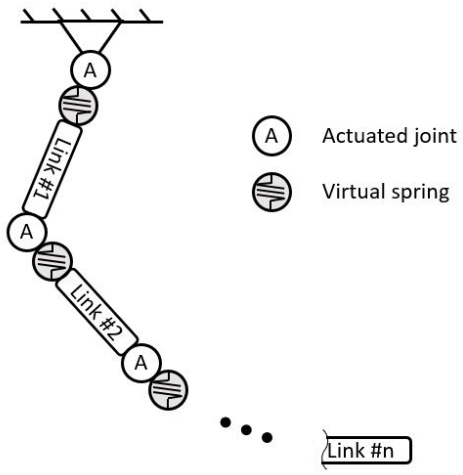

FIGURE 5. VJM model for stiffness.

For our approach we use the VJM approach where the virtual springs attached between joints and links are considering both the joint stiffness and link stiffness. The stiffness for the whole robotic manipulator is calculated by

$$\boldsymbol{K}_c = diag(K_x, K_y, K_z, K_{Rx}, K_{Ry}, K_{Rz}) \quad (12)$$

Each of these elements are calculated as follows:

$$K_x = \frac{1}{\sum_{i=1}^{n_{DOF}} K_{x,J_i}^{-1} + K_{x,L_i}^{-1}} \quad (13)$$

where K_x is the stiffness at the end effector in the current pose when subjected to a force in the $\mathbf{x} = [1, 0, 0]^T$ direction. K_{x,J_i} is the stiffness of joint i when the end effector is subjected to a force in the $\mathbf{x}$ direction in the current pose.

K_{x,L_i} is the stiffness of link i when the end effector is subjected to a force in the $\mathbf{x}$ direction in the current pose. The stiffness

$$K_{x,J_i} = \frac{1}{K_{x,J_i,tilt}^{-1} + K_{x,J_i,tors}^{-1}} \tag{14}$$

dependent on the tilting stiffness $K_{x,J_i,tilt}$ of joint i and the torsional stiffness $K_{x,J_i,tors}$ of joint i which can be calculated by

$$K_{x,J_i,tilt} = \frac{K_{J_i,tilt}}{||\mathbf{M}_{x,i} \times \mathbf{z}_{J_i}||^2} \tag{15}$$

where $K_{J_i,tilt}$ is the stiffness rating of joint i in the tilt axis and

$$\mathbf{M}_{x,i} = \mathbf{P}_{EE}^{i} \times \mathbf{x} \tag{16}$$

is the vector describing the moment working on joint i when $\mathbf{x} = [1, 0, 0]^T$ and $\mathbf{P}_{EE}^{i}$ is a vector from the center of the joint to the end effector of the robot in the current pose, with respect to the world frame. $\mathbf{z}_{J_i}$ is the unit vector representing the rotational axis of the joint in the current robot configuration. Similarly for the torsional stiffness of joints:

$$K_{x,J_i,tors} = \frac{K_{J_i,tors}}{||\mathbf{M}_{x,i} \cdot \mathbf{z}_{J_i}||^2} \tag{17}$$

The stiffness of links are calculated by

$$K_{x,L_i} = \frac{1}{K_{x,L_i,tilt}^{-1} + K_{x,L_i,tors}^{-1}} \tag{18}$$

where

$$K_{x,L_i,tilt} = \frac{K_{L_i,tilt}}{||\mathbf{M}_{x,i} \times \mathbf{z}_{L_i}||^2} \tag{19}$$

and

$$K_{x,L_i,tors} = \frac{K_{L_i,tors}}{||\mathbf{M}_{x,i} \cdot \mathbf{z}_{L_i}||^2} \tag{20}$$

is the tilting stiffness and torsional stiffness from link i when a force is applied to the in the $\mathbf{x}$ direction at the end effector, respectively.

$$\mathbf{Z}_{L_i} = \mathbf{P}_{i+1}^{i} \tag{21}$$

is the vector from center of joint i to the next. In the case of the last link, it is the vector from the last joint to the End Effector. $\mathbf{M}_{x,i}$ is the same as in Eq. 16.

E. BUCKLING OF CYLINDRICAL BEAMS

According to Brazier [23], the max moment on a cylindrical shell before buckling can be expressed as:

$$M = \frac{2\sqrt{2}}{9} \frac{E\pi rt^2}{\sqrt{1-\nu^2}} \approx 1.035Ert^2 \tag{22}$$

where E is the Young's modulus, r is the radius and t is the thickness.

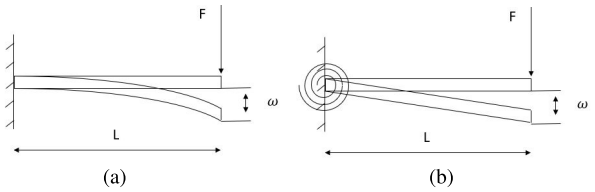

FIGURE 6. Flexing of a cantilever beam when a load is applied at the free end. (a) Flexible beam (b) Virtual rigid beam with spring.

F. STIFFNESS OF CYLINDRICAL BEAMS AS CANTILEVERS

Conventional beam theory states that a cantilever beam of length L with end load F has a deflection given by

$$\omega = \frac{FL^3}{3EI} \tag{23}$$

where the area moment of inertia I for a hollow cylinder is

$$I = \frac{\pi}{4}({r_{OD}}^4 - {r_{ID}}^4) \tag{24}$$

The stiffness can then be expressed as a spring coefficient:

$$K = \frac{F}{\omega} = \frac{3EI}{L^3} \tag{25}$$

where the unit for K is $[N/m]$. The spring coefficient can also be represented by $[Nm/rad]$:

$$K_{ang} = \frac{3EI}{L} \tag{26}$$

G. TORSIONAL STIFFNESS OF CYLINDRICAL BEAMS

Torsional stiffness of a cylindrical beam of length L can be described in terms of the angular deflection

$$\theta = \frac{LT}{JG} \tag{27}$$

where T is the applied torque, G is the modulus of rigidity and the polar moment of inertia J for a hollow cylinder is

$$J = \frac{\pi}{2}({r_{OD}}^4 - {r_{ID}}^4) \tag{28}$$

The stiffness can then be expressed as a spring coefficient:

$$K_\theta = \frac{T}{\theta} = \frac{JG}{L} \tag{29}$$

H. END-PLATES

The end-plates are for the purpose of the optimization assumed to be infinitely stiff, with a length of 0.05 meters. Additionally, the offset from the joint attachment point to the centerline of the cylindrical portion of the links (resulting in torsional torque on the cylinder) is assumed to be so small that it can be neglected.

IV. KINEMATICS

The position and rotational pose of the end effector of a serial link robot can be described as a series of transformations from the previous joint to the current joint. The transformations are given by

$$[\mathbf{H}] = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \tag{30}$$

where **R** is the rotation matrix, **p** is a vector which represents the displacement, **0** is the zero vector.

A widely used approach to setting up transformation between joints is the DH convention. The DH parameters can be extracted from the robot and the corresponding transformation calculated. The transformation using the DH convention can be done as follows:

$$\left[\mathbf{H}\right] = \left[\text{Trans}_z(d) \cdot \text{Rot}_z(\theta) \cdot \text{Trans}_x(a) \cdot \text{Rot}_x(\alpha)\right] \quad (31)$$

where d is the offset between the common normal along the previous z, θ is the angle between the old x to the new x, about the previous z, a is the length of the common normal, and α is the angle from old z axis to new z axis, about the common normal. The y-axis is completed with the right hand rule. However, the DH convention does not necessarily place the center of the joint at the correct location, even though the transformation is correct. An example is shown in Figure 7.

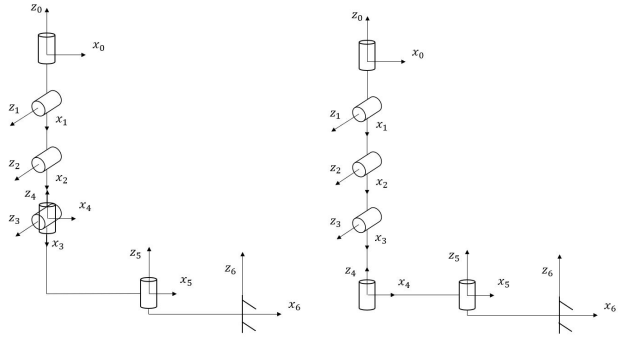

FIGURE 7. Comparison of the kinematic representations of the 6DOF manipulator when using DH and when not using it. Not drawn to scale. (a) The kinematics when using the DH convention. (b) The correct kinematics when considering the joint centers.

Multiplying all the link transformations then gives the total transformation from the base of the robot to the end effector. This is called the forward kinematics, where the joint values are known, and the end effector pose is calculated. The opposite problem, finding the joint values with a given end effector pose, is called the inverse kinematics.

V. OPTIMIZATION USING GA

A. GENETIC ALGORITHM

The Genetic Algorithm is a heuristic method inspired by the natural selection found in nature, and belongs to the larger class of evolutionary algorithms. Genetic algorithms became popular in the 1970s by the book Adaptation in Natural and Artificial systems by John Holland [24]. A series of candidate solutions with random attributes are initially generated. This is called a population. The individuals of the population are then evaluated with the given cost/fitness function. The population is then evolved into (hopefully) a better population. This evolution process mimics nature, by breeding, mutation and survival of the fittest. A flow chart of the GA is shown in Figure 8.

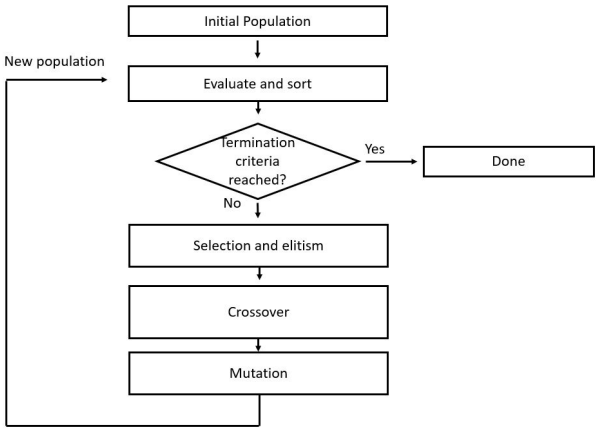

FIGURE 8. Flow chart of the GA process.

TABLE 2. Stiffness, max torque and weight of the joints.

Joint	Max Tilting Torque [Nm]	Max Torsional Torque [Nm]	Tilting Rigidity [Nm/rad]	Torsional Rigidity [Nm/rad]	Weight [kg]
1	2000	1000	$6.30 * 10^5$	$1.03 * 10^6$	30.0
2	3280	1750	$4.99 * 10^6$	$2.23 * 10^6$	45.5
3	1355	675	$1.92 * 10^6$	$7.04 * 10^5$	21.8
4	550	260	$1.89 * 10^6$	$3.26 * 10^5$	10.0
5	400	180	$7.73 * 10^5$	$1.44 * 10^5$	6.5
6	800	100	$2.86 * 10^5$	$3.43 * 10^5$	1.5

B. NORMALIZATION

A key aspect of a GA is that is can be applied to a wide variety of problems without much change of the internal structure. One of the techniques that enables this is normalization. For a continuous GA, each of the genes inside a chromosome can only contain values between 0 and 1. When testing the chromosomes, each gene has to be un-normalized in order to evaluate the fitness of that individual.

C. EXPLOITATION

Two variations of the GA are used in these experiments - a standard GA (STD), and a GA with extra exploitation (EXP). Both of the GA variation uses elitism, i.e. saving a set amount of the best candidate solutions. In the EXP GA, for each elite candidate solution, a number of new individuals are generated. These individuals are based on the elite individuals, but with a random number between -0.1 and 0.1 added to each of the elite individuals chromosome. This increases the exploitation of the genetic algorithm.

D. EXPERIMENTAL SETUP

The test was done with all the joints chosen beforehand. The stiffness, maximum torque and weight of the joints can be seen in Table 2.

The experiment was implemented using the statics explained in Section III. For the experiment, it was assumed that the COG of the joint were in the middle of the kinematic joint, and the COG of the links were in the middle of the link length. The material for the links in the experiment are AISI 304.

In total 12 parameters were to be optimized: the four link lengths marked in Table 1, the diameter of these links, and the thickness of these links. The boundaries of the 12 parameters can be seen in Table 3. For simplicity, the boundaries are divided into length, diameter and thickness.

TABLE 3. Parameter boundaries.

	Link length [mm]	Link diameter [mm]	Link thickness [mm]
Min	200	10	0.1
Max	1700	250	10

TABLE 4. Parameters of the GA.

Population size	50
Nr of Elite	3
Nr of Near Searches pr Elite	4
Mutation Rate	0.5
Mutation Gain	0.1
Evolution iterations	200

The parameters of the GA in the experiment can be seen in Table 4. The manipulator DH table can be seen in Table 1.

In order to test the candidate solutions, a test set of robot configurations had to be generated. For our experiment, only a worst case test case was used, where the robot is at full horizontal reach in the $\mathbf{x} = [1, 0, 0]^T$ direction. However, for other optimization problems, it might be desirable to build a set of test cases where the robot is set to operate within a part of the working envelope. In our experiment the multiple objectives are put into a single objective function in order to produce a single solution instead of a Pareto front.

E. TEST CASES

Two cases for the fitness function was set up. For the first case, the fitness function was formulated to maximize reach, with a minimum payload capability. The minimum payload was set at 100N on the end effector and a minimum reach of 2 meters was set as a requirement.

Case 1:

$$F = F_P + F_R + F_S - W \tag{32}$$

where

$$F_P = \begin{cases} 100, & \text{if } P > 100. \\ -100, & \text{otherwise.} \end{cases} \tag{33}$$

is the fitness with respect to payload (P) $[N]$,

$$F_R = \begin{cases} 100R, & \text{if } R > 2. \\ -1000, & \text{otherwise.} \end{cases} \tag{34}$$

is the fitness with respect to reach (R) $[m]$,

$$F_S = \begin{cases} 0.001S, & \text{if } S > 4000. \\ -1000, & \text{otherwise.} \end{cases} \tag{35}$$

is the fitness with respect to the average of the stiffnesses in the different directions (S) $[N/m]$ and W is weight $[kg]$.

In the second case, the most important aspect is the payload of the robot. A minimum reach is set at 1 meter, while stiffness and weight is also considered.

Case 2:

$$F = F_P + F_R + F_S - W \tag{36}$$

where

$$F_P = \begin{cases} P, & \text{if } P > 100. \\ -100, & \text{otherwise.} \end{cases} \tag{37}$$

$$F_R = \begin{cases} 100, & \text{if } R > 1. \\ -1000, & \text{otherwise.} \end{cases} \tag{38}$$

$$F_S = \begin{cases} 0.001S, & \text{if } S > 4000. \\ -1000, & \text{otherwise.} \end{cases} \tag{39}$$

In both cases, both the standard GA (STD) and the GA with the extra exploitation (EXP) was used.

For our application, the stiffness at the end effector is not of high concern. Spraying results in a relatively low force, and some deflection (<1cm) does not significantly decrease the performance of the robot for the given task. However, a minimum value for the stiffness is set up the same way as for the payload and reach to force the GA to optimize the stiffness to some extent.

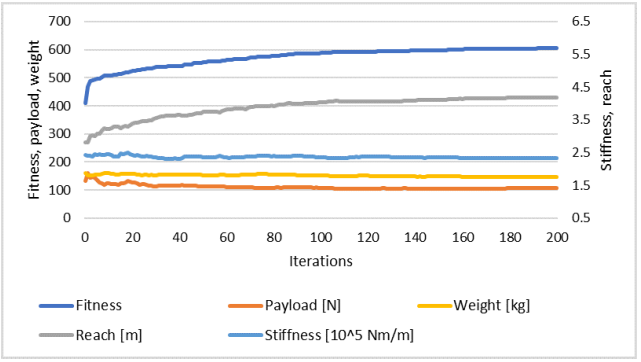

FIGURE 9. Evolution of fitness and robotic manipulator properties vs iteration for case 1, using standard GA (STD). Average of 20 runs.

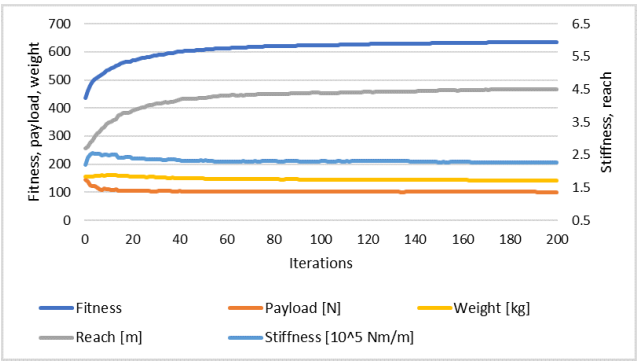

FIGURE 10. Evolution of fitness and robotic manipulator properties vs iteration for case 1, using GA with extra exploitation (EXP). Average of 20 runs.

F. RESULT

The optimization process was performed 20 times. The best candidates of the GA from the 20 runs for the two cases are shown in Table 6. A chart showing the average evolution of the fitness and the properties of the solution for case 1 from the 20 runs found by the GA are shown in Figure 9 and 10.

TABLE 5. Parameters of the initial design.

Link	Link diameter [mm]	Link thickness [mm]
1	150	1.0
2	150	1.0
3	50	1.0
4	50	1.0

TABLE 6. Unnormalized chromosomes of the best candidates from 20 runs.

Case	GA	Link	Length [m]	Diam. [mm]	Thickn. [mm]
1	STD	1	1.303	245.5	1.4
		2	0.797	218.0	2.2
		3	0.475	134.6	2.3
		4	1.700	237.1	0.1
	EXP	1	1.459	239.7	1.2
		2	0.933	250.0	0.5
		3	0.512	182.6	0.5
		4	1.700	239.2	0.1
2	STD	1	0.248	127.0	6.1
		2	0.200	196.7	4.2
		3	0.200	214.5	1.0
		4	0.200	250.0	0.3
	EXP	1	0.244	197.9	3.2
		2	0.200	250.0	2.0
		3	0.200	135.2	2.0
		4	0.200	169.8	0.1

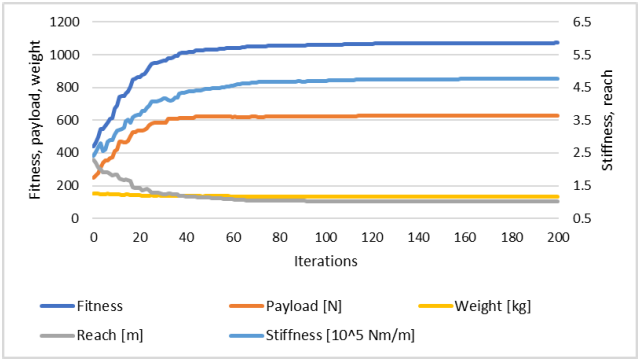

FIGURE 11. Evolution of fitness and robotic manipulator properties vs iteration for case 2, using standard GA (STD). Average of 20 runs.

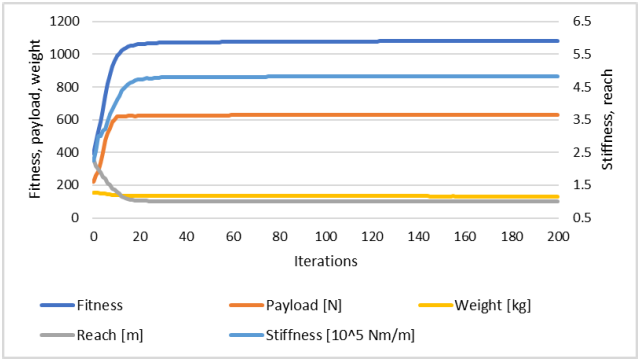

FIGURE 12. Evolution of fitness and robotic manipulator properties vs iteration for case 2, using GA with extra exploitation (EXP). Average of 20 runs.

Case 2 are shown in Figure 11 and 12. It is interesting to note the smoother and quicker convergence of the GA with the extra exploitation, for both cases. For the fitness functions used in the two cases it can be assumed that there exist only one optimal solution, and thus the exploration trade-off is acceptable. Additionally, the GA with extra exploitation finds more optimal solutions, as shown in Table 7.

TABLE 7. Characteristics of the initial design and the candidates from Table 6. Stiffness at full reach in x.

Characteristic	Init. design	Case 1		Case 2	
GA type		STD	EXP	STD	EXP
Fitness		623.52	646.96	1080.66	1083.93
Max payload [N]	88.63	100.76	100.23	630.12	630.86
Reach [m]	3.72	4.31	4.63	1.00	1.00
Weight [kg]	136.68	149.27	140.85	132.07	130.82
Stiffn. in **x** [N/m]	324427	709031	658625	787001	786775
Stiffn. in **y** [N/m]	10241	8593	7490	307287	309398
Stiffn. in **z** [N/m]	9538	8076	6995	353541	355479

Comparing the initial design with the optimized design of case 1 with extra exploitation, the payload is increased with 13.1% while the reach is increased by 24.5%, with an increase in weight of 3.1%. However, there has also been a decrease in stiffness. This is due to the fitness function not emphasizing the stiffness.

Taking the best candidates as basis, the robot manipulator will have the characteristics for the two cases with the different algorithms as described in Table 7. For reference, the characteristics of the initial design is added. The lengths are given in Table 1, the link diameters and thicknesses can be seen in Table 5. The program used for the optimization can be downloaded from https://github.com/NTNU-IHB/RobotOptGA. The average computation time for 20 runs of 200 iterations was 71 seconds for a computer with a Intel Core i7 processor.

VI. CONCLUSION AND FUTURE WORK

The GA turned out to be a good tool for optimizing design for a given criteria. Notice how close the optimized design is to the optimizing criteria, e.g 100.76 and 100.23 payload, when the fitness function wanted 100.00 for case 1, using standard GA and GA with extra exploitation, respectively, and a reach of 1.00 meters when the fitness function wanted 1.00 for case 2, for both algorithms.

Using GA for optimizing robotic manipulator design can be applied to more complex considerations than what has been done in this research without much trouble. For instance, for a real world application, it might be suitable to only have a set of different link dimensions to choose from. In addition, the GA can be used to optimize robotic manipulators regardless of number of DOFs, and can handle both prismatic and revolute joints.

The limitations of the DH convention and the joint stiffness matrix become obvious in this work. Since the COG of the joints were assumed to be in the kinematic joint center, the DH convention could not be used to achieve the kinematics for the manipulator. The joint stiffness matrix does not give a complete description of the stiffness of a manipulator, thus a complete description for the stiffness using the VJM approach had to be developed.

The GA could also be used to optimize the payload of the robotic manipulator inside a part of the work envelope, simply by using a different test set in the evaluation algorithm. Precision of the manipulator has not been directly considered in this work. However, precision of a manipulator is to some

extent linked to the stiffness of the manipulator, which has been considered in this work.

Future work will include building a prototype which is optimized by a given fitness function using the methodology outlined in this paper. The prototype will be used to verify the performance improvement offered by the methodology.

CONFLICT OF INTEREST

The author declares that there is no conflict of interest regarding the publication of this article.

REFERENCES

[1] *World Robotics*. Int. Fed. Robot., Frankfurt, Germany, 2013.

[2] C. Zieliński *et al.*, "Variable structure robot control systems: The RAPP approach," *Robot. Auton. Syst.*, vol. 94, pp. 226–244, Aug. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889016306248

[3] Y. Sun, H. Liu, Z. Luo, and F. Wang, "Robot mechanical structure optimization design," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2007, pp. 1919–1923.

[4] T. Kivelä, J. Mattila, and J. Puura, "A generic method to optimize a redundant serial robotic manipulator's structure," *Autom. Construct.*, vol. 81, pp. 172–179, Sep. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580517305150

[5] O. Chocron and P. Bidaud, "Genetic design of 3D modular manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, Apr. 1997, pp. 223–228.

[6] W. K. Chung, J. Han, Y. Youm, and S. H. Kim, "Task based design of modular robot manipulator using efficient genetic algorithm," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, Apr. 1997, pp. 507–512.

[7] A. Jafari, M. Safavi, and A. Fadaei, "A genetic algorithm to optimum dynamic performance of industrial robots in the conceptual design phase," in *Proc. IEEE 10th Int. Conf. Rehabil. Robot.*, Jun. 2007, pp. 1129–1135.

[8] C. West, A. Montazeri, S. D. Monk, and C. J. Taylor, "A genetic algorithm approach for parameter optimization of a 7DOF robotic manipulator," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1261–1266, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896316309648

[9] Y. Tu, G. Yang, Q. Cai, L. Wang, and X. Zhou, "Optimal design of SINS's Stewart platform bumper for restoration accuracy based on genetic algorithm," *Mechanism Mach. Theory*, vol. 124, pp. 42–54, Jun. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094114X17317457

[10] R. Wang and X. Zhang, "Optimal design of a planar parallel 3-DOF nanopositioner with multi-objective," *Mech. Mach. Theory*, vol. 112, pp. 61–83, Jun. 2017.

[11] E. Francalanza, A. Fenech, and P. Cutajar, "Generative design in the development of a robotic manipulator," *Procedia CIRP*, vol. 67, pp. 244–249, Jun. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212827117311514

[12] R. Saravanan, S. Ramabalan, N. G. R. Ebenezer, and C. Dharmaraja, "Evolutionary multi criteria design optimization of robot grippers," *Appl. Soft Comput.*, vol. 9, no. 1, pp. 159–172, Jun. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494608000525

[13] P. J. Bentley and J. P. Wakefield, "Conceptual evolutionary design by a genetic algorithm," *Eng. Des. Autom.*, vol. 3, no. 2, pp. 119–131, Jun. 1997. [Online]. Available: http://eprints.hud.ac.uk/id/eprint/3941/

[14] R. Kelaiaia, O. Company, and A. Zaatri, "Multiobjective optimization of a linear Delta parallel robot," *Mechanism Mach. Theory*, vol. 50, pp. 159–178, Apr. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094114X11002126

[15] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," Eidgenössische Technische Hochschule Zürich, Inst. Technische Informatik Kommunikationsnetze, Zurich, Switzerland, Tech. Rep. 103, 2001.

[16] R. Bye, O. Osen, and B. S. Pedersen, "A computer-automated design tool for intelligent virtual prototyping of offshore cranes," *Proc. 29th Eur. Council Modelling Simulation (ECMS)*, 2015, pp. 147–156.

[17] E. Bjørlykhaug, L. Giske, T. Løvdal, O. J. Mork, and O. Egeland, "Development and validation of robotic cleaning system for fish processing plants," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–6.

[18] J.-J. Park, B.-S. Kim, J.-B. Song, and H.-S. Kim, "Safe link mechanism based on nonlinear stiffness for collision safety," *Mechanism Mach. Theory*, vol. 43, no. 10, pp. 1332–1348, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094114X07001577

[19] J. Angeles and F. C. Park, "Performance Evaluation and Design Criteria," in *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2008, pp. 229–244, doi: 10.1007/978-3-540-30301-5_11.

[20] J. K. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," in *Proc. 19th IEEE Conf. Decis. Control Including Symp. Adapt. Processes*, Dec. 1980, pp. 95–100.

[21] S.-F. Chen and I. Kao, "Conservative congruence transformation for joint and cartesian stiffness matrices of robotic hands and fingers," *Int. J. Robot. Res.*, vol. 19, no. 9, pp. 835–847, 2000, doi: 10.1177/02783640022067201.

[22] A. Pashkevich, A. Klimchik, and D. Chablat, "Enhanced stiffness modeling of manipulators with passive joints," *Mechanism Mach. Theory*, vol. 46, no. 5, pp. 662–679, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094114X10002338

[23] L. G. Brazier, "On the flexure of thin cylindrical shells and other 'thin' sections," *Proc. Roy. Soc. London A, Math. Phys. Sci.*, vol. 116, no. 773, pp. 104–114, 1927. [Online]. Available: http://rspa.royalsocietypublishing.org/content/116/773/104

[24] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

EMIL BJØRLYKHAUG is currently pursuing the Ph.D. degree with the Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology. His current research focus is robotic technologies for automating fish processing, tools that may facilitate industrial robots in performing more complex tasks and artificial intelligence.

OLAV EGELAND received the Siv.Ing. and Dr. Ing. degrees in electrical engineering from the Norwegian University of Science and Technology (NTNU) in 1984 and 1987, respectively. He was a Professor in robotics with the Department of Engineering Cybernetics, NTNU, from 1989 to 2004. He was the co-founder, CTO, and CEO of Marine Cybernetics AS from 2004 to 2011. He is currently a Professor of Production Automation with the Department of Mechanical and Industrial Engineering, NTNU. His research interests are within modeling, simulation and control of mechanical systems, with applications to robotics and off-shore systems.