Structural Optimisation and Design of a Cable-Driven Hyper-Redundant Manipulator for Confined Semi-Structured Environments

Rami Al-Khulaidi *, Rini Akmeliawati, Steven Grainger and Tien-Fu Lu

Robotics and Automation Research Group, School of Mechanical Engineering, The University of Adelaide, Adelaide, SA 5005, Australia
* Correspondence: a1776973@adelaide.edu.au

Abstract: Structural optimisation of robotic manipulators is critical for any manipulator used in confined semi-structured environments, such as in agriculture. Many robotic manipulators utilised in semi-structured environments retain the same characteristics and dimensions as those used in fully-structured industrial environments, which have been proven to experience low dexterity and singularity issues in challenging environments due to their structural limitations. When implemented in environments other than fully-structured industrial environments, conventional manipulators are liable to singularity, joint limits and workspace obstacles. This makes them inapplicable in confined semi-structured environments, as they lack the flexibility to operate dexterously in such challenging environments. In this paper, structural optimisation of a hyper-redundant cable-driven manipulator is proposed to improve its performance in semi-structured and challenging confined spaces, such as in agricultural settings. The optimisation of the manipulator design is performed in terms of its manipulability and kinematics. The lengths of the links and the joint angles are optimised to minimise any error between the actual and desired position/orientation of the end-effector in a confined semi-structured task space, as well as to provide optimal flexibility for the manipulators to generate different joint configurations for obstacle avoidance in confined environments. The results of the optimisation suggest that the use of a redundant manipulator with rigid short links can result in performance with higher dexterity in confined, semi-structured environments, such as agricultural greenhouses.

Keywords: robotics in agriculture; hyper-redundant; redundant manipulators; confined environments; semi-structured environments; structural optimisation

1. Introduction

The introduction of robotics in the middle of the twentieth century promoted the use of robots in many industrial and commercial automated contexts [1–5]. However, they are mostly used in fully-structured industrial environments [6,7], where the robot workspaces are clearly defined and known a priori. Only recently have robots been further utilised in semi-structured environments [8–11], such as in agriculture, in which the surroundings are dynamic and can be uncertain. Some work has presented strategies for introducing robotic manipulators into such environments [8,10,12,13]. Nevertheless, all the manipulators used are similar to conventional manipulators, with the 5-6 degrees of freedom (DOFs) that are found in a fully-structured industrial environment. Such robots have been proven to be inefficient, as the manipulators suffer from a lack of dexterity appropriate for the given workspace [10,14,15].

A semi-structured environment is defined as an environment with known landmarks that has no static topology, where the surroundings cannot be classified into, for example, rooms and walls [16]. It is instead a dynamic environment and subject to change during the operation of the manipulators [17]. This means any interactions with the environment
require continuous replanning of the path for the end-effector to reach a certain point [18]. As robotic manipulators interact with dynamic environments, they should be sufficiently flexible to achieve any given task in their workspace. Conversely, a structured environment is an environment in which the manipulators repeat the same task in predefined and fully controlled regions [19,20]. An example of a conventional robotic manipulator that has been tested in a semi-structured environment is a bespoke platform developed to harvest lettuce [14]. The platform consists of a vision system with a convolutional neural network and custom end-effector software. The vision system shows a reasonable performance. However, the performance of the 6-DOF conventional robotic manipulator has proven to be inefficient due to its structural limitations, specifically the length of its links. As a result, although 69 lettuces could be detected by the vision system, only 31 lettuces were successfully harvested from the 60 that were within the range of the manipulator. This is due to the length of the links that connect every two consecutive joints, which proved too long to have the required dexterity to operate in the given workspace. This shows that conventional robotic manipulators are unable to provide the high degree of flexibility needed to work in semi-structured environments, which further raises the need for highly-dexterous manipulators that can move freely and flexibly within their workspace. This flexibility can be achieved using redundant/hyper-redundant manipulators, which have more degrees of freedom than are required to perform specific tasks in the workspace [21,22]. The inverse kinematics of such manipulators have an infinite number of solutions. This can be beneficial in confined semi-structured/unstructured environments, as the redundancy can be used for an infinite number of configurations without affecting the end-effector location [23].

The movement of manipulators in a confined semi-structured environment is determined by the positioning of the end-effector, which is the primary task, and the manipulator’s configuration, which is the movement of all joints, as a secondary task. Thus, in addition to the primary task (end-effector positioning), the redundancy of redundant/hyper-redundant manipulators is used to perform a secondary task, which changes the manipulator’s configuration in confined and enclosed spaces, such as in agriculture, construction, tooling, inspection, space exploration and underwater applications [24–26].

Robotic manipulators in confined semi-structured environments are typically required to perform tasks that are relatively complex, such as picking crops or fruit [27]. These environments constrain manipulators from moving freely due to dimensional limitations and the presence of obstacles, such as tree branches. To overcome the challenges and enable robotic manipulators to move freely and flexibly without damaging or hitting tree branches, fruit, or any other obstacles, structural optimisation should be conducted.

Some recent work on the structural optimisation of robotic manipulators has been presented to optimise the structure of serial and parallel manipulators by finding optimal link lengths [28,29]. For example, Bjorlykhaug et al. optimised the structure of a 6-DOF serial manipulator using Genetic Algorithms (GA) [30]. A generic method was proposed to optimise the structure of a redundant serial manipulator [31]; however, kinematics optimisation was only performed to evaluate the positioning of the end-effector, without considering other important measures, such as dexterity and manipulability.

Manipulability is critical for high dexterity [32]. Dexterity can be defined as the ability of the manipulator to reach a specific location in the workspace [33]. While the manipulability of a robot is its capacity to change the end effector’s position as a function of the joint configuration [34]. Manipulability is measured using a manipulability index [35–37], and the greater the manipulability index, the more dexterous the robot. The manipulability index value is affected by the number of joints and the lengths of the links. As dexterity has diverse definitions, in this article we define dexterity in semi-structured environments as the ability of the manipulator to reach a specific point using different configurations. Therefore, in the case of semi-structured environments, dexterity is highly dependent on manipulability because manipulability reveals if the manipulator can perform different configurations flexibly to reach a specific point.
Before implementing the robotic manipulators in semi-structured environments, their dexterity and manipulability must be considered in the optimisation process to ensure appropriacy and applicability. In the earlier work on semi-structured environments, the manipulability criteria, despite their critical role, have not been considered. The research to date only considered conventional industrial manipulators, which are not applicable in confined semi-structured environments due to the long distance between joints. The distance between every two consecutive joints should be optimised for better manipulability in confined semi-structured environments. Therefore, this article aims at extending the aforementioned works by optimising the structure of hyper-redundant manipulators to enable them to work flexibly with high manipulability within their given workspace in confined semi-structured environments. The introduced optimisation methods can result in an optimal design of robotic manipulators as the manipulability measures are considered in the optimisation process. This makes the optimised manipulator highly applicable in such confined environments as the manipulability measures are critical in terms of assessing the performance of robotic manipulators in this type of environment.

In this study, the robotic arm’s joint angles and link lengths are optimised using three manipulability measures and inverse kinematics optimisation to ensure the manipulator can move flexibly within its workspace with different configurations. Two algorithms are considered for the optimisation: Artificial Bees Colony (ABC) [38] and Fmincon [39], which is a built-in optimisation algorithm in MATLAB. Then, the optimisation results are validated using a hyper-redundant manipulator, which was created based on the optimisation results. The performance of the developed manipulator is tested in terms of manipulability by generating a collision-free trajectory in a confined semi-structured environment and analysing its manipulability at certain points within the environment. The results show that the optimised manipulator can achieve high manipulability even though the environment used for the validation is different from those used for optimisation. Based on the optimisation results, this implies that the manipulator can be used in any given confined semi-structured environment as it has high manipulability measures.

The remaining sections of this article are organised as follows: Section 2 presents the structural optimisation of a hyper-redundant manipulator to find the optimal length of links and joint angles that enable the manipulator to move with high manipulability. Section 3 shows the results of the proposed optimisation methods, along with the validation of the results. The conclusion is drawn in Section 4.

2. Materials and Methods
2.1. Structural Optimisation of Hyper-Redundant Manipulators

Optimisation is the process of determining the optimal solution among diverse alternatives; it is necessary for robotic manipulator design so that the optimal structure of joint parameters can be obtained. Several optimisation algorithms are available, including Genetic Algorithms (GA), Particle Swarm Optimisation (PSO) and Artificial Bees Colony (ABC). No optimisation algorithm can be classified as the unqualified best, as the performance of each specific algorithm depends on the problem structure, which means some objective functions can perform faster than others, depending on the problem. However, ABC generally showed successful and stable results when compared with GA and PSO when it was tested to minimise the makespan for job scheduling, and protein production problems [40,41], as it showed advantages in stability and not being trapped in local minima. Moreover, it does not employ external parameters such as crossover and mutation rates, as in GA [42]. Another widely used optimisation solver is the Fmincon solver; it is the main optimisation solver in the MATLAB Optimisation Toolbox [39], which performs better for nonlinearly constrained problems. Fmincon is based on interior point techniques for nonlinear minimisation. The flowcharts of both optimisation algorithms are shown in Figures 1 and 2.
techniques for nonlinear minimisation. The flowcharts of both optimisation algorithms are shown in Figure 1 and 2.

Figure 1. Flowchart of the Artificial Bees Colony algorithm.

Figure 2. Flowchart of the Fmincon optimisation algorithm.
Any optimisation problem can be defined as finding the vector of parameters that minimises or maximises an objective function:

$$\min \text{ or } \max f(z) = (z_1, z_2, \ldots, z_n)$$  

and constrained by:

$$z_{\text{min}} \leq z_i \leq z_{\text{max}}, \; i = 1, 2, \ldots, n$$

where $z$ represents the vector consisting of parameters to be optimized. In this study, $z_i$ is defined as $x_i = [q_i, a_i]^T$, where $q = (q_1, q_2, \ldots, q_n)$ is the joint angles in degrees, and $a$ is the link length of a hyper-redundant manipulator in mm. In this case, all the links are assumed to have an equal length. The objective function used in this study to optimise the joint parameters for $n$ DOF is given by:

$$\begin{cases}
\max f(x), \\
x = [q, a], \\
\text{s.t. } q_i \leq q \leq \bar{q}, \\
a \leq a \leq \bar{a}, \\
i = 1, 2, \ldots, n
\end{cases}$$  

where $f(.)$ is the manipulability measure, which is used as the objective function, and $q_i, \bar{q}$ and $a, \bar{a}$ are the lower bound and upper bound for joint angles and the length of the link, respectively. When choosing lower and upper bound values, the joint structure and limitations were considered, as well as the designs in previous studies that showed a lack of flexibility to work in their given workspace [14,24,43]. Thus, the lower bound was chosen to avoid being too small to provide sufficient space for other components, such as pulleys, wires and joint sensors, and the upper bound was sufficiently large that its size would not affect its manipulability. That is, if the optimized length of the link were too small, pulleys, wires and sensors cannot be added to the joint due to space limitations; while, if it is too long, it will result in a manipulator that is not flexible enough to handle its workspace as with those in the aforementioned studies.

For the proposed redundant manipulator, in order to achieve high manipulability and dexterity, the optimised objective functions are based on manipulability measures and inverse kinematics. Based on the definitions of dexterity and manipulability, the objective functions are dependent on the joint parameters. The minimum and maximum values used in both optimisation algorithms are $\pm 90$ degrees for joint angles and $80$ mm and $250$ mm for the length of the link, respectively. The minimum value is chosen, as mentioned above, to provide sufficient space for the joint components, whilst the dexterity performance is used to define the maximum value as the link length, for which greater than $250$ mm was proven to lack dexterity in any workspace [14].

The manipulator includes universal joints, as described further in Section 3. Thus, every joint is driven by two motors—one for each axis. The joint is attached to two pulleys that are connected to the motors by cables to enable each motor to drive one axis. The last joint is considered a general end-effector so that it can be incorporated into any type of bespoke gripper. Another aspect of the design is the base, which houses the motors, electronics and power equipment. The cables that are used in the design are of the Bowden cable type and are assumed in this work to have no effect on the kinematics and dynamics.

### 2.2. Optimisation Algorithms

The first optimisation algorithm used in our work to determine the optimal joint parameters is ABC as it can avoid being trapped in local minima wherein different joint parameters are tested and compared. ABC is a metaheuristic method inspired by foraging honeybees. There are three groups of bees in ABC: employed bees, onlooker bees and scout bees. Onlooker bees and scout bees are also called unemployed bees. Initially, the scout bees look for food until they find its source. Then, the employed bees and onlooker
bees exploit the abundance (fitness) of the food source. The employed bees are linked to a certain food source and share information about the food by means of dancing. Their dancing is observed by onlooker bees to determine the best source using probability. The location of the food source reflects a solution in the ABC algorithm, and the abundance of food represents the fitness of the solution.

For the proposed structural optimisation, the initialisation stage of the algorithm starts by assigning initial values to the joint angles and length of the link. Then, it initialises the population and the limits of the variables. Next, it starts by assigning the initial solution, which is, in this study, zero for manipulability optimisation, with infinity as the distance between the current and desired end-effector coordinates. The corresponding position for every employed bee represents the values of the joint parameters. Abundance is shown by the manipulability index and the distance between the current and desired position of the end-effector.

After the initialisation process, the employed bees start searching for a new food source \( v_m \), which is more abundant than \( x_m \), which is already in their memory, and evaluate its \( i \)-th fitness using the following equation [38]:

\[
v_{mi} = x_{mi} + \phi_{mi}(x_{mi} - x_{ki})
\] (3)

where \( i \) is a randomly selected parameter index, \( x_k \) is a randomly chosen food source, and \( \phi_{mi} \) is chosen randomly between \(-1\) and \(1\). Then, after the new food source is identified using Equation (3), the fitness of the new food source is calculated, and a greedy selection is performed between \( v_{mi} \) and \( x_{mi} \), as follows:

\[
fit_m(x_m) = \begin{cases} 
1 & f_m(x_m) \geq 0 \\
1 + \text{abs}(f_m(x_m)) & f_m(x_m) < 0
\end{cases}
\] (4)

where \( fit_m(x_m) \) is the fitness of the new food source, and \( f_m(x_m) \) is the objective function value for the solution \( x_m \). Then, the information is shared by the employed bees with the onlooker bees, which determine the best solution among the fitness values using probability, \( p_m \). The best solution is determined using the roulette wheel fitness selection method [44], as in Equation (5):

\[
p_m = \frac{f_m(x_m)}{\sum_{m=1}^{SP} f_m(x_m)}
\] (5)

where \( SP \) is the size of the population, which was chosen, based on several trials, to be 500, divided equally into employed and onlooker bees so the employed bees share the information about their food source to recruit the corresponding onlooker bees. Based on the shared information, the onlooker bees choose the food source. Then, the employed bees, whose solution cannot be improved after certain trials, abandon their solution using abandonment criteria. As a result, they become scout bees, looking for a new source of food. The abandonment criteria define the control limit for jumping from local minima to global minima; that is, if the solution of the employed bees does not change after a number of iterations, they are converted to scout bees. The overall flowchart of the algorithm is shown in Figure 1.

The number of iterations in ABC is 300 for manipulability measures and 1000 for inverse kinematics. The objective function values for manipulability measures did not increase from around 250 iterations, which means the best value is reached at around 250 iterations. This was observed after several trials when testing the algorithm. Therefore, the number of iterations was reduced to 300 to save computational time. In Fmincon, on the other hand, the initial number of iterations was set to 5000, and the exit criteria depend on the tolerance, which is \(1e^{-16}\). This means that although the number of iterations was set too high, the optimisation can stop after a few iterations as long as the objective function value is not increasing, as the results show in the next section.
If, for example, the algorithm starts by assigning the number of joint angles, based on the given length of the manipulator workspace, the length of the links is identified. The next step is identifying the distance between every two consecutive joints at each iteration. Then, the algorithm starts by assigning the minimum number of joints and checking the manipulability and dexterity with iteratively assigned joint parameter values and storing the value of the corresponding objective function. Next, it assigns a different number of joint parameters and evaluates their objective function and compares the value of the current objective function with the previous objective function value. As the manipulability depends on the value of the specific joint angles, the subroutine has to iteratively assign different angles and check the manipulability measures.

Finally, after considering the optimisation results, a computer-aided design (CAD) model of the manipulator will be created. Every link should have two axes (pitch and yaw) so the manipulator can move freely and flexibly and cover the area within the range of the manipulator. As highlighted previously, the main reason for the low level of dexterity is the long length of the links. In other words, every consecutive similar motion (for example, yaw-yaw) should be close to one another, based on the optimised values of each joint’s parameters, so the manipulator will be sufficiently dexterous for the specific task.

2.3. Kinematics of Hyper Redundant Manipulators

The optimisation in this study is based on inverse kinematics and manipulability. The different manipulability measures used for optimisation are based on the Jacobian matrix. The kinematics of the proposed hyper-redundant manipulator are solved using Denavit-Hartenberg (DH) parameters, as shown in Table 1. The model consists of $n$ revolute joints along the $z$-axis of frames $\{1\}$ to $\{n\}$. As the manipulator is a universal joint-type manipulator, $n$ is an even number. The forward kinematics function, $h(\cdot)$, is:

$$y = h(q)$$

where $y = (x, y, z, \theta_x, \theta_y, \theta_z)^T$ is the pose vector (the translational pose in the $x$, $y$ and $z$ axes, and the rotational angles with respect to the $x$, $y$ and $z$ axes) in the task space, and $q = (q_1, q_2, \ldots, q_{n-1}, q_n)^T$ is the joint angle vector in the configuration space.

Table 1. DH parameters.

| Joint | $q_n$ | $d_n$ | $a_n$ (mm) | $\alpha_n$ (deg) |
|-------|-------|-------|------------|-----------------|
| 1     | $q_1$ | $d_1$ | $a_1$      | $\alpha_1$      |
| 2     | $q_2$ | $d_2$ | $a_2$      | $\alpha_2$      |
| \ldots | \ldots | \ldots | \ldots | \ldots   |
| $n-1$ | $q_{n-1}$ | $d_{n-1}$ | $a_{n-1}$ | $\alpha_{n-1}$ |
| $n$   | $q_n$ | $d_n$ | $a_n$      | $\alpha_n$      |

The transformation between the base frame and any other frame is given by the transformation matrix as:

$$T = 
\begin{pmatrix}
\cos(q_n) - \cos(a_n) \sin(q_n) & \sin(a_n) \sin(q_n) & a_n \cos(q_n) \\
\sin(q_n) \cos(a_n) & -\sin(q_n) \cos(a_n) & a_n \sin(q_n) \\
0 & \sin(a_n) & \cos(a_n) \\
0 & 0 & 0
\end{pmatrix}$$

where $d_n$ is the offset along the $Z_{n-1}$ axis; $a_n$ is the length of the common normal, which in our case is the length of the link connecting the previous yaw movement axis to the next pitch movement axis; and $\alpha_n$ is the angle about the common normal, from the $Z_{n-1}$ axis to the $Z_n$ axis.
The inverse kinematics are used to find the function \( h^{-1} \), which maps the task space to the configuration space:

\[
q = h^{-1}(y)
\]  

(8)

The configuration space of this type of manipulator is larger than the task space, which results in infinite inverse kinematics solutions with \( v \) number of equations and \( u \) unknowns, where \( v > u \), with \( u = 6 \), which corresponds to 3 translation and 3 angular motions in a 3D space. Therefore, iterative methods using the Jacobian matrix are used, as analytical solutions do not exist. Deriving Equation (6) with respect to time gives:

\[
\dot{y} = J\dot{q}
\]

(9)

where \( J \) is the Jacobian matrix that maps the translational and the angular velocities of the joints to the pose velocities.

\[
J = \begin{pmatrix}
\frac{\partial y_1}{\partial q_1} & \cdots & \frac{\partial y_1}{\partial q_v} \\
\vdots & \ddots & \vdots \\
\frac{\partial y_u}{\partial q_1} & \cdots & \frac{\partial y_u}{\partial q_v}
\end{pmatrix}
\]

(10)

As the inverse of the Jacobian matrix can only exist for non-singular square matrices, the Moore-Penrose pseudoinverse is taken to give:

\[
\dot{q} = J^+\dot{y}
\]

(11)

where \( J^+ \) is the pseudoinverse of the Jacobian matrix, which is equal to:

\[
J^+ = J^T(JJ^T)^{-1}
\]

(12)

where \( J^T \) is the transpose of the Jacobian matrix.

3. Results and Discussion

3.1. Manipulability Optimisation

Manipulability optimisation has been undertaken to ensure a high manipulability measure. Three different manipulability measures were used as objective functions to optimise the best joint parameter values. The first measure used is the Yoshikawa manipulability measure [32], which is given by:

\[
\mu = \sqrt{\det(JJ^T)}
\]

(13)

where \( \det(JJ^T) \) is the determinant of the multiplication of the Jacobian matrix and its transpose. The Yoshikawa measure is equal to the product of the singular values of the Jacobian matrix. As this index is proportional to the velocity ellipsoid, the high singular values mean a high end-effector velocity for the unit norm: a manipulator is considered to have a high manipulability if its velocity ellipsoid is isotropic, i.e., has a sphere-shape, which means it can change its end-effector position in all three axes equally. The objective function values and the optimised joint parameters for the optimisation based on the Yoshikawa manipulability measure using Fmincon and ABC are shown in Figures 3 and 4.

However, the Yoshikawa measure is not an accurate manipulability measure, as the velocity ellipsoid cannot depict the exact status of every singular value. For example, if one singular value is very small, the velocity ellipsoid can still be large as the other singular values are high. Furthermore, this measure is unbounded: therefore, another measure was used to compensate for the downside of the Yoshikawa manipulability measure: the Local Conditioning Index (LCI). The LCI is the ratio of the minimum and maximum singular values [45,46], Equation (14):

\[
\text{LCI} = \frac{\sigma_{\text{min}}}{\sigma_{\text{max}}} \in [0,1]
\]

(14)
where $\sigma_{\text{min}}$ and $\sigma_{\text{max}}$ are the minimum and maximum singular values, respectively.

Figure 3. Objective function values (left), and the corresponding optimised joint parameters (right) for the best objective function value obtained by Fmincon for the Yoshikawa manipulability measure.

Figure 4. Objective function values (left), and the corresponding optimised joint parameters (right) for the best objective function value obtained by ABC for the Yoshikawa manipulability measure.

The result obtained from the LCI manipulability measure as an objective function for parameter optimisation is shown in Figures 5 and 6:

Figure 5. Objective function values (left), and the corresponding optimised joint parameters (right) for the best objective function value obtained by Fmincon for the LCI manipulability measure.
The third manipulability measure that was used to optimise the parameters is a new measure introduced by Mariass et al. [47]. This measure is bounded and can be written as a function of the joint coordinates. The measure is given by:

\[
\Delta H = \frac{H}{A} \in [0, 1]
\]  

\[
H = \frac{u}{\text{tr}\left((J^T J)^{-1}\right)}
\]  

\[
A = \frac{\text{tr}(J J^T)}{u}
\]

where \( H \) and \( A \) are the harmonic and arithmetic mean of eigenvalues, respectively, and \( \text{tr} \) is the trace of the matrix. Figures 7 and 8 show the result for the \( \Delta H \) measure when used as an objective function for parameter optimisation.

It can be seen from the results of the three manipulability measures that the optimisation algorithms continued to reduce the links’ length until it reached the lower bound limit of length between every two joints, resulting in a 16-DOF manipulator. This means the shorter the length of links, the better the manipulability that can be achieved. However, in practice, the length of the link should be sufficient to avoid design limitations by providing space for joint sensors, cables and pulleys. Hence, the lower bound of the link lengths is restricted to 80 mm.
3.2. Inverse Kinematics (IK) Optimisation

Inverse kinematics optimisation was conducted to verify the performance of the manipulator, considering certain task space coordinates. It is based on the results from the manipulability optimisation in Section 2.2 in terms of the optimal number of joints. The objective function and constraints for joint parameters in this part are similar to Equation (2), with $G(\ldots)$, Equation (18), as the objective function, which minimises the error between the current and desired positions of the end-effector. Additional workspace constraints were added to ensure the manipulator can perform flexibly in confined and enclosed workspaces. The workspace was constrained to resemble a box with dimensions of 1000 mm, 700 mm and 900 mm in the $x$, $y$ and $z$ coordinates, respectively. This means the end-effector should not go beyond this workspace. Another constraint was added to limit the middle joint to spaces no larger than 500 mm, 350 mm and 450 mm in the $x$, $y$ and $z$ coordinates, respectively.

$$G(x, y, z) = \sqrt{(x - x_d)^2 + (y - y_d)^2 + (z - z_d)^2}$$  \hspace{1cm} (18)

where $x$ and $x_d$ are the current and desired end-effector positions in the $x$-axis, respectively. Similarly, $y$ and $y_d$, and $z$ and $z_d$ are the current and desired end-effector coordinates in the $y$-axis and $z$-axis, respectively. The desired coordinates were 500 mm, 300 mm and 200 mm in the $x$, $y$ and $z$ coordinates, respectively. The result obtained by Fmincon for inverse kinematics optimisation was more accurate than the result obtained by ABC, as the error was nearly 0.002 mm using the Fmincon solver, while it was about 0.4 mm in ABC, as shown in Figures 9 and 10:

**Figure 9.** Objective function values (left), and the corresponding optimised joint parameters (right) for the best objective function value obtained by Fmincon for the IK optimisations.
Figure 10. Objective function values (left), and the corresponding optimised joint parameters (right) for the best objective function value obtained by ABC for the IK optimisations.

All the optimised values of parameters obtained by the different objective functions used in this study are summarised in Table 2.

Table 2. The optimal joint parameters corresponding to the best objective functions.

| Joint Parameters | Manipulability Measures | Inverse Kinematics Optimisation |
|------------------|-------------------------|---------------------------------|
|                  |                        | Fmincon | ABC | Fmincon | ABC | Fmincon | ABC | Fmincon | ABC |
|                  |                        | 7.4 × 10^8 | 1.5 × 10^9 | 0.0101 | 0.0107 | 4.3 × 10^{-4} | 6.8 × 10^{-4} | 4.14 × 10^{-6} | 0.46 |
| Objective Function Value | q1 (deg) | 11.4 | −2.22 | 71 | 90.0 | 79.49 | 24.2 | 30.5 | 35 |
|                    | q2 (deg) | 29.3 | −1 | −12 | −90.0 | −3.67 | 87.1 | 25.6 | 32 |
|                    | q3 (deg) | 0 | 33.6 | −88.7 | −90.0 | −89.91 | 17.8 | 14.6 | −2 |
|                    | q4 (deg) | 9.3 | 8.91 | −84.9 | −20.6 | −61.2 | 90.0 | −17 | −0.3 |
|                    | q5 (deg) | 0 | −5.49 | −35 | −90.0 | −52.3 | 19.5 | −16 | −80 |
|                    | q6 (deg) | −9.2 | −12.6 | 56.8 | −3.83 | 66 | 90.0 | −19 | 16 |
|                    | q7 (deg) | 0 | 2.02 | 46.6 | −89.2 | 34.7 | 83.8 | 54 | 67 |
|                    | q8 (deg) | −12.2 | 1.89 | 80.3 | −21.2 | 65.8 | 73.8 | 13.8 | −61 |
|                    | q9 (deg) | 90 | −90.0 | −17 | 6.3 | −13 | 41.4 | 65.6 | 67 |
|                    | q10 (deg) | −90 | 3.62 | 57.5 | 50.7 | 62 | −22.8 | 78 | 70 |
|                    | q11 (deg) | −9.8 | 0.487 | −39.4 | −90.0 | −35.5 | −33.6 | 70.5 | −14.6 |
|                    | q12 (deg) | 0 | 2.32 | 27.7 | −38.6 | 37 | −90.0 | 71.6 | 0.5 |
|                    | q13 (deg) | −4.7 | 13.3 | 30.8 | −88.9 | 48.4 | −5.19 | −8.8 | −68 |
|                    | q14 (deg) | 0 | 7.36 | 85.18 | 39.5 | 81 | −88.5 | 68 | −38 |
|                    | q15 (deg) | −29 | 38.4 | 10.7 | −90.0 | 27.3 | −90.0 | −37 | −90 |
|                    | q16 (deg) | 90 | 3.27 | 89.67 | −9.72 | 90 | −90.0 | 17 | −90 |
|                     | a (mm) | 120.00 | 150.0 | 100 | 90.0 | 100.0 | 90.1 | 107 | 124 |
Based on the results of the different objective functions, either from manipulability measures or inverse kinematics, it is observable that the best performance of the manipulator can be obtained within a range between 90 mm and 150 mm for the length of the links. The joint angles also reached the lower and upper bounds of ±90 degrees. Therefore, considering the obtained results, the manipulator designed in this work has a length of 100 mm between every two consecutive similar joints, which satisfies the high dexterity needs of the manipulator.

3.3. Results Validation

Based on the optimisation results, a model for the optimised manipulator was developed using Autodesk Inventor, as shown in Figure 11. It consists of the housing for the actuators and electronic equipment, along with the manipulator. To account for dexterity, the universal joint is considered in the design, as shown in Figure 11 (right). The joint consists of two identical yokes joined by an octagon in the centre. Every yoke is connected to a pulley so it can perform two-axis (pitch and yaw) movements. The distance between every two consecutive joints is 100 mm, and the distance from the last joint to the tip is also 100 mm, as shown in Figure 12; the inner diameter of the joint is 60 mm, while its outer diameter is 90 mm. The joints’ movements are animated, and they can move ±90 degrees in both the pitch and the yaw axes.

![Figure 11. The overall structure of the manipulator (left) and the design of the joint (right).](image1)

![Figure 12. The last three joints’ structure.](image2)
To evaluate the dexterity and manipulability of the proposed manipulator, a confined semi-structured environment was created, as shown in Figure 13. It is a different environment from the environment used in the optimisation process to show that the manipulator, based on the optimisation results, can perform effectively in any confined semi-structured environment. The dimensions of the environment are 1.2 m in the x-axis, 0.6 m in the y-axis and 0.55 m in the z-axis. The environment consists of various shapes of obstacles including two boxes, two spheres and a cylinder. The first box is located at 0.4 m, 0.2 m and 0.2 m, in the x, y and z axes, respectively, while the second box is placed at 0.9 in the x-axis, 0.2 in the y-axis and 0.1 in the z-axis. The first box has a length of 0.2 m, a width of 0.1 m and a height of 0.15 m, while the second box is 0.3 m, 0.3 m and 0.08 m in length, width and height, respectively. Two spheres with a radius of 0.08 m were placed in the environment. The first one was positioned at [0.6 0.35 0.5] in the [x y z] axes, while the second one was at [0.6 0.35 0.5] in the [x y z] axes. The cylinder had a radius of 0.05 m and a length of 0.25 m; it was located at 0.75, 0.1 and 0.45 in the x, y and z axes, respectively.

Figure 12. The last three joints' structure.

To evaluate the dexterity and manipulability of the proposed manipulator, a confined semi-structured environment was created, as shown in Figure 13. It is a different environment from the environment used in the optimisation process to show that the manipulator, based on the optimisation results, can perform effectively in any confined semi-structured environment. The dimensions of the environment are 1.2 m in the x-axis, 0.6 m in the y-axis and 0.55 m in the z-axis. The environment consists of various shapes of obstacles including two boxes, two spheres and a cylinder. The first box is located at 0.4 m, 0.2 m and 0.2 m, in the x, y and z axes, respectively, while the second box is placed at 0.9 in the x-axis, 0.2 in the y-axis and 0.1 in the z-axis. The first box has a length of 0.2 m, a width of 0.1 m and a height of 0.15 m, while the second box is 0.3 m, 0.3 m and 0.08 m in length, width and height, respectively. Two spheres with a radius of 0.08 m were placed in the environment. The first one was positioned at [0.6 0.35 0.5] in the [x y z] axes, while the second one was at [0.6 0.35 0.5] in the [x y z] axes. The cylinder had a radius of 0.05 m and a length of 0.25 m; it was located at 0.75, 0.1 and 0.45 in the x, y and z axes, respectively.

Figure 13. Enclosed semi-structured environments with the obstacles’ coordinates (left) and the manipulator’s home configuration (right).

Next, a collision-free trajectory was generated to drive the manipulator from the home configuration, Figure 13 (right), at an end-effector pose of 1.026 m, 0 m and 0 m at the x, y and z axes respectively, to a final pose of 0.803 m in the x-axis, 0.154 m in the y-axis and 0.398 m in the z-axis, which is a location that is slightly behind the cylinder. Four waypoints were generated between the home configuration and the final pose. The middle way point was behind the second sphere at coordinates of 0.66 m in the x-axis, 0.29 m in the y-axis and 0.47 m in the z-axis. The middle waypoint and final pose configurations of the manipulator are reflected in Figure 14. The joint position and velocity trajectories are shown in Figure 15. Considering the restrictions of the confined workspace, the joint positions did not go to their maximum limit. The highest joint position was for joint one with 52.4 degrees, followed by joints nine, seven and eight with 40.7, 35.5 and −34.4 degrees, respectively; while joints four and eleven did not go above −28 degrees. Seven joints (two, six, ten, thirteen, fourteen, fifteen and sixteen) did not move at all, while joints three, five and twelve remained at almost zero throughout the trajectory. Considering the various obstacles in the given environment, the manipulator was able to move around obstacles and reach its goal without using the ±90-degrees full range of angles. Thus, implementing the manipulators resulting from the optimisation used in this study in environments with a different set of obstacles has been shown to result in a flexible performance given the high manipulability results.
the confined workspace, the joint restrictions imposed by the confined workspace. Therefore, the manipulator, with its design its end-effector positions in all axes, using different configurations of joints, despite the both isotropic (have a spherical shape). This implies that the manipulator can change that the proposed manipulator has a high manipulability as the velocity ellipsoids are the final pose.

Figure 14. The manipulator configuration at the middle waypoint (left) and the final pose (right).

Figure 15. Joint angle trajectory positions (left) and velocity (right) from the home configuration to the final pose.

Figure 16 shows the velocity ellipsoids for the manipulator, with the design based on the optimisation results, at the middle and final pose configurations. The figure shows that the proposed manipulator has a high manipulability as the velocity ellipsoids are both isotropic (have a spherical shape). This implies that the manipulator can change its end-effector positions in all axes, using different configurations of joints, despite the restrictions imposed by the confined workspace. Therefore, the manipulator, with its design being based on the optimisation results, can move freely and generate high velocity in the x, y and z axes, regardless of the shape of the obstacles.
Figure 16. Velocity ellipsoid of the joint angles corresponding to the middle waypoint configuration (left) and final pose configuration (right).

4. Conclusions

This paper proposed a concept for structural optimisation of hyper-redundant manipulators that can be used in confined semi-structured environments, such as in agriculture. Two methods of optimisation were used, using the Fmincon and Artificial Bees Colony (ABC) algorithms, considering the manipulability measures and inverse kinematics, to ensure optimal joint parameters that enable the manipulator to work flexibly within its task space.

This optimisation method was conducted for manipulators with universal joints, as the universal joints provide a wide range of motion for the manipulator to cover two axes with a 360-degree workspace, so all points within the manipulator workspace are reachable. The optimised manipulator was assumed to have a 1 m length, and the optimisation method was conducted to ensure high performance within its workspace, so the reachability of the points within its workspace with the optimised optimal lengths of the links was shown to be successful. Therefore, for long manipulators, an additional number of links alone need to be added to the manipulator for it to cover its workspace dexterously.

A hyper-redundant manipulator was designed based on the results of optimisation to be used for validation in a confined semi-structured environment. Its performance was tested by assessing its manipulability in certain confined locations. It shows a high level of dexterity using different joint configurations, as it is depicted by manipulability ellipsoids. Unlike previous studies that only optimised conventional manipulators, which considered kinematic constraints alone, the optimisation in this study depends on the manipulability measures and inverse kinematics in a confined workspace. This shows the manipulator is dexterous and performs with high manipulability within any given workspace, as is shown as the testing environment was completely different from that in the optimisation scenario. This is demonstrated by the results of the validation, as the manipulator was able to reach its goal and move around the different obstacles with a range of angles between ±60 degrees, which proves the high level of flexibility of the optimised manipulator in handling its workspace. Therefore, more complicated workspaces with different obstacles can be handled within the optimised manipulator as its range of angles is ±90 degrees. This means the manipulator can perform flexibly within any given confined semi-structured environment or workspace. To the best of our knowledge, this is the first work that optimises hyper-redundant manipulators in semi-structured environments. The previous works only consider conventional manipulators, and dexterity and manipulability measures have been neglected in their optimisation. As a future extension of this study,
various control techniques will be investigated to ensure the accuracy and robustness of the manipulator’s performance in semi-structured environments.

Author Contributions: Conceptualization, R.A.-K., R.A., S.G. and T.-F.L.; Data curation, R.A.-K.; Formal analysis, R.A.-K., R.A., S.G. and T.-F.L.; Methodology, R.A.-K.; Software, R.A.-K.; Supervision, R.A., S.G. and T.-F.L.; Validation, R.A.-K.; Writing—original draft, R.A.-K.; Writing—review & editing, R.A., S.G. and T.-F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Nuzzi, R.; Brusasco, L. State of the art of robotic surgery related to vision: Brain and eye applications of newly available devices. Eye Brain 2018, 10, 13–24. [CrossRef] [PubMed]
2. Moghaddam, B.M.; Chhabra, R. On the guidance, navigation and control of in-orbit space robotic missions: A survey and prospective vision. Acta Astronaut. 2021, 184, 70–100. [CrossRef]
3. Cianchetti, M.; Laschi, C.; Menciassi, A.; Dario, P. Biomedical applications of soft robotics. Nat. Rev. Mater. 2018, 3, 143–153. [CrossRef]
4. Al-Khulaidi, R.A.; Akmeliawati, R.; Azlan, N.Z.; Bakr, N.H.A.; Fauzi, N.M. Development of robotic hands of signbot, advanced Malaysian sign-language performing robot. Adv. Robot. Res. 2018, 2, 183. [CrossRef]
5. Kim, M.J.; Chi, H.-L.; Wang, X.; Ding, L. Automation and Robotics in Construction and Civil Engineering. J. Intell. Robot. Syst. 2015, 79, 347–350. [CrossRef]
6. Oyekan, J.O.; Hutabarat, W.; Tiwari, A.; Grech, R.; Aung, M.H.; Mariani, M.P.; López-Dávalos, L.; Ricaud, T.; Singb, S.; Dupuis, C. The effectiveness of virtual environments in developing collaborative strategies between industrial robots and humans. Robot. Comput.-Integr. Manuf. 2019, 55, 41–54. [CrossRef]
7. Domae, Y. Recent trends in the research of industrial robots and future outlook. J. Robot. Mechatron. 2019, 31, 57–62. [CrossRef]
8. Bock, T. The future of construction automation: Technological disruption and the upcoming ubiquity of robotics. Autom. Constr. 2015, 59, 113–121. [CrossRef]
9. Oesterreich, T.D.; Teuteberg, F. Understanding the implications of digitisation and automation in the context of Industry 4.0: A triangulation approach and elements of a research agenda for the construction industry. Comput. Ind. 2016, 83, 121–139. [CrossRef]
10. Bac, C.W.; Hemming, J.; van Tuijl, B.A.J.; Barth, R.; Wais, E.; van Henten, E.J. Performance Evaluation of a Harvesting Robot for Sweet Pepper. J. Field Robot. 2017, 34, 1123–1139. [CrossRef]
11. Kurita, H.; Iida, M.; Cho, W.; Suguri, M. Rice Autonomous Harvesting: Operation Framework. J. Field Robot. 2017, 34, 1084–1099. [CrossRef]
12. Silwal, A.; Davidson, J.R.; Karkee, M.; Mo, C.; Zhang, Q.; Lewis, K. Design, integration, and field evaluation of a robotic apple harvester. J. Field Robot. 2017, 34, 1140–1159. [CrossRef]
13. Botterill, T.; Paulin, S.; Green, R.; Williams, S.; Lin, J.; Saxton, V.; Mills, S.; Chen, X.; Corbett-Davies, S. A Robot System for Pruning Grape Vines. J. Field Robot. 2017, 34, 1100–1122. [CrossRef]
14. Birrell, S.; Hughes, J.; Cai, J.Y.; Iida, F. A field-tested robotic harvesting system for iceberg lettuce. J. Field Robot. 2018, 37, 225–245. [CrossRef] [PubMed]
15. Funami, Y.; Kawakura, S.; Tadano, K. Development of a Robotic Arm for Automated Harvesting of Asparagus. Eur. J. Agric. Food Sci. 2020, 2, 1–10. [CrossRef]
16. Liu, M.; Colas, F.; Oth, L.; Siegwart, R.Y. Incremental topological segmentation for semi-structured environments using discretized GVG. Auton. Robot. 2015, 38, 143–160. [CrossRef]
17. Meng, Y.; Sun, Y.; Chang, W.-S. Optimal trajectory planning of complicated robotic timber joints based on particle swarm optimization and an adaptive genetic algorithm. Constr. Robot. 2021, 5, 131–146. [CrossRef]
18. Zhang, L.; Wang, Y.; Zhao, X.; Zhao, P.; He, L. Time-optimal trajectory planning of serial manipulator based on adaptive cuckoo search algorithm. J. Mech. Sci. Technol. 2021, 35, 3171–3181. [CrossRef]
19. Huang, J.; Hu, P.; Wu, K.; Zeng, M. Optimal time-jerk trajectory planning for industrial robots. Mech. Mach. Theory 2018, 121, 530–544. [CrossRef]
20. Rout, A.; Deepak, B.; Biswal, B.B.; Mahanta, G.B. Trajectory Generation of an Industrial Robot With Constrained Kinematic and Dynamic Variations for Improving Positional Accuracy. Int. J. Appl. Metaheuristic Comput. 2021, 12, 163–179. [CrossRef]
21. Nho Cho, C.; Jung, H.; Son, J.; Gi Kim, K. A modular control scheme for hyper-redundant robots. *Int. J. Adv. Robot. Syst.* 2015, 12, 91. [CrossRef]

22. Siciliano, B.; Khatib, O.; Kröger, T. *Springer Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 200.

23. Miteva, L.; Chavdarov, I.; Yovchev, K. Trajectory Planning for Redundant Robotic Manipulators with Constrained Joint Space. In *Proceedings of the 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, 17–19 September 2020; pp. 1–6.

24. Tang, L.; Wang, J.; Zheng, Y.; Gu, G.; Zhu, L.; Zhu, X. Design of a cable-driven hyper-redundant robot with experimental validation. *Int. J. Adv. Robot. Syst.* 2017, 14, 1729881417734458. [CrossRef]

25. Tang, J.; Zhang, Y.; Huang, F.; Li, J.; Chen, Z.; Song, W.; Zhu, S.; Gu, J. Design and Kinematic Control of the Cable-Driven Hyper-Redundant Manipulator for Potential Underwater Applications. *Appl. Sci.* 2019, 9, 1142. [CrossRef]

26. Xu, W.; Liu, T.; Li, Y. Kinematics, Dynamics, and Control of a Cable-Driven Hyper-Redundant Manipulator. *IEEE/ASME Trans. Mechatron.* 2018, 23, 1693–1704. [CrossRef]

27. Bac, C.W.; van Henten, E.J.; Hemming, J.; Edan, Y. Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead. *J. Field Robot.* 2014, 31, 888–911. [CrossRef]

28. Pierrot, F.; Nabat, V.; Company, O.; Krut, S.; Poignet, P. Optimal Design of a 4-DOF Parallel Manipulator: From Academia to Industry. *IEEE Trans. Robot.* 2009, 25, 213–224. [CrossRef]

29. Sun, Y.; Liu, H.; Luo, Z.; Wang, F. Robot mechanical structure optimization design. In *Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Sanya, China, 15–18 December 2007; pp. 1919–1923.

30. Bjørlykhaug, E.; Egeland, O. Mechanical Design Optimization of a 6DOF Serial Manipulator Using Genetic Algorithm. *IEEE Access.* 2018, 6, 59087–59095. [CrossRef]

31. Kivelä, T.; Mattila, J.; Puura, J. A generic method to optimize a redundant serial robotic manipulator’s structure. *Autom. Constr.* 2017, 81, 172–179. [CrossRef]

32. Yoshikawa, T. Manipulability of Robotic Mechanisms. *Int. J. Robot. Res.* 1985, 4, 3–9. [CrossRef]

33. Klein, C.A.; Blaho, B.E. Dexterity Measures for the Design and Control of Kinematically Redundant Manipulators. *Int. J. Robot. Res.* 1987, 6, 72–83. [CrossRef]

34. Jin, L.; Li, S.; La, H.M.; Luo, X. Manipulability Optimization of Redundant Manipulators Using Dynamic Neural Networks. *IEEE Trans. Ind. Electron.* 2017, 64, 4710–4720. [CrossRef]

35. Chandrashekhar, A. Generation of Manipulability Ellipsoids for Different Configurations Using the Yoshikawa’s Manipulability Index and Manipulability Ellipsoid. *Int. J. Theor. Appl. Mech.* 2017, 12, 623–635.

36. Dufour, K.; Suleiman, W. On Maximizing Manipulability Index while Solving a Kinematics Task. *J. Intell. Robot. Syst.* 2020, 100, 3–13. [CrossRef]

37. Dufour, K.; Suleiman, W. On integrating manipulability index into inverse kinematics solver. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 24–28 September 2017; pp. 6967–6972.

38. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Erciyes University: Kayseri, Turkey, 2005.

39. MathWorks Fmincon. Available online: https://au.mathworks.com/help/optim/ug/fmincon.html (accessed on 23 July 2020).

40. Muthiah, A.; Rajkumar, R. A Comparison of Artificial Bee Colony algorithm and Genetic Algorithm to Minimize the Makespan for Job Shop Scheduling. *Procedia Eng.* 2014, 97, 1745–1754.

41. Alqattan, Z.N.M.; Abdullah, R. A Comparison between Artificial Bee Colony and Particle Swarm Optimization Algorithms for Protein Structure Prediction Problem; Springer: Berlin/Heidelberg, Germany, 2013; pp. 331–340.

42. Rao, R.S.; Narasimham, S.; Ramalingaraju, M. Optimization of Distribution Network Configuration for Loss Reduction Using Artificial Bee Colony Algorithm. *Int. J. Electr. Comput. Eng.* 2008, 2, 1964–1970.

43. Marais, W.; Göktoğan, A. Design and Control of CRAM: A Highly Articulated Cable-driven Remote Access Manipulator for Confined Space Inspection. In *Proceedings of the Australasian Conference on Robotics and Automation (ACRA-2017)*, Sydney, Australia, 11–13 December 2017.

44. Karaboga, D. Artificial bee colony algorithm. *Scholarpedia* 2010, 5, 6915. [CrossRef]

45. Mohammadi, H.R.; Zsombor-murray, P.J.D.; Angeles, J. The Isotropically Design of Two General Classes of Planar Parallel Manipulators. *J. Robot. Syst.* 1995, 12, 795–805. [CrossRef]

46. Hosseini, M.A.; Daniali, H.M. Weighted local conditioning index of a positioning and orienting parallel manipulator. *Sci. Iran.* 2011, 18, 115–120. [CrossRef]

47. Marais, W.; Göktoğan, A. A New Manipulability Measure for the Control of CRAM: A Cable-driven Remote Access Manipulator. In *Proceedings of the Australasian Conference on Robotics and Automation (ACRA-2017)*, Sydney, Australia, 11–13 December 2017.