

TEL AVIV UNIVERSITY

School of Mechanical Engineering

Mechanical Engineering

**OPTIMAL KINEMATIC
DESIGN OF MANIPULATOR
TO DETECT EARLY STRESSES
IN GREENHOUSE CROPS**

A thesis submitted toward the degree of

MSc in Mechanical Engineering

By

Tamir Mhabary

This research was carried out in The School of
Mechanical Engineering

Under the supervision of

Pr. Avital Bechar

Dr. Amiram Moshaiov

May 2020

TEL AVIV UNIVERSITY

School of Mechanical Engineering
Mechanical Engineering

**OPTIMAL KINEMATIC
DESIGN OF MANIPULATOR
TO DETECT EARLY STRESSES
IN GREENHOUSE CROPS**

A thesis submitted toward the degree of
MSc in Mechanical Engineering
By

Tamir Mhabary

This research was carried out in The School of
Mechanical Engineering

Under the supervision of

Pr. Avital Bechar

Dr. Amiram Moshaiov

May 2020

Acknowledgments

I wish to express my deepest gratitude to my advisors, Prof. Avital Bechar and Dr. Amiram Moshaiov for their insight, support, guidance, and assistance.

I would like to thank BARD (US-ISRAEL BINATIONAL AGRICULTURAL RESEARCH AND DEVELOPMENT FUND) for funding the project, without their funding, the research could not be conducted.

Great thanks to all the members in the Agricultural Robotics Lab (ARL) for their support and help when needed.

Abstract

Table of Contents

1	Introduction	1
2	Literature Review	2
2.1	Robotics	2
2.1.1	Robot	2
2.1.2	Robotics in Agriculture	2
2.1.3	Robotics Manipulators.....	3
2.2	Optimization	5
2.2.1	Evolutionary Algorithm.....	5
2.2.2	Pareto Front	6
2.2.3	Window Of Interest	7
2.2.4	set-based concept	8
2.3	Manipulators Optimization.....	8
3	Methods and materials	9
3.1	Manipulator arms.....	9
3.1.1	Definition	9
3.1.2	Parameters.....	9
3.1.3	Assumptions.....	10
3.1.4	Indices of manipulator performance.....	10
3.2	Simulation	11
3.2.1	Simulator.....	11
3.3	Optimization	12
3.3.1	the set-based concept approach.....	12
3.3.2	Objectives.....	13
3.3.3	Independent variables:.....	13
3.3.4	Constrains	13
3.3.5	Concepts	14
3.4	An interface between the simulator and optimization algorithms	15
3.4.1	Configuration builder.....	15
3.4.2	Indices Calculator	15
3.5	Preliminary Window of Interest.....	16
3.5.1	Configuration selection.....	16
3.5.2	Pareto Front	16

3.6. Evolutionary optimization with Dynamic-Window of interest.....	16
3.6.1. Evolution Strategies.....	16
3.6.2. Resource Allocation	20
3.7. Find Pareto front using a multi-objective evolutionary algorithm	21
4. Results	22
4.1. Manipulator arms.....	22
4.1.1. Assumptions.....	22
4.1.2. Indices of manipulator performance.....	23
4.2. Simulation	25
4.2.1. Simulator	25
4.3. Preliminary Window of Interest.....	25
4.4. Evolutionary optimization with Dynamic-Window of interest.....	27
4.4.1. Resource Allocation – Greedy Method	27
4.4.2. Resource Allocation – Fair Method	28
5. Conclusions and future work.....	29
6. Bibliography.....	30

LIST OF FIGURES

FIGURE 1 - BASIC EA	6
FIGURE 2 - PARETO FRONT & DOMINANCE[21]	7
FIGURE 3 - PARETO FRONT DTLZ1[22]	7
FIGURE 4 - CONFIGURATION EXAMPLE	9
FIGURE 5 - MANIPULATOR DETECTION POINTS	10
FIGURE 6 – MOTOMAN(LEFT) & RIGHT UR5(RIGHT)	12
FIGURE 7 - INTERFACE BETWEEN OPTIMIZATION TO THE SIMULATOR	15
FIGURE 8 - EVOLUTION STRATEGIES	17
FIGURE 9 - ALL JOINTS & AXES POSSIBILITIES (3 DOF)	22
FIGURE 10 - ALL JOINTS & AXES POSSIBILITIES ONE BRANCH	23
FIGURE 11 - LCI VS MANIPULABILITY INDEX	24
FIGURE 12 - MID-PROXIMITY JOINT INDEX VS MANIPULABILITY INDEX	24
FIGURE 13 - MANIPULABILITY INDEX VS DOF	24
FIGURE 14 - TIME COMPASSION	25
FIGURE 15 - ACCUMULATED WEIGHT VS ACCUMULATED LENGTH	25
FIGURE 16 - PRELIMINARY WOI	26
FIGURE 17 - GREEDY METHOD - PARAMETERS CHECK	28

LIST OF TABLES

TABLE 1 - SELECTED INDICES	11
TABLE 2 - CONCEPTS	14
TABLE 3 - PARTS ID	19
TABLE 4 - ID AND ITS NEIGHBORS	20
TABLE 5 - CONCEPTS IN WOI	26
TABLE 6 - CONFIGURATIONS IN WOI	27
TABLE 7 - CHECKED CONCEPTS	27

1. Introduction

A major challenge in greenhouse crops is the inability to detect stresses and risks early enough preventing the uncontrolled spreading of stresses causing irreparable damage. Often, although the knowledge of how to handle stress is available, it is too late to act correctly due to late detection. Hence, farmers often react too wastefully. Therefore, there is a compelling need to develop an effective, affordable robotic inspection system using close sensing. In greenhouse environments, the conditions are specially controlled to maximize the crops growth rate and production, which can expose plants to biotic and abiotic risks. Due to scarce human resources, time limitations, and the high cost of current monitoring methods, mostly manual inspection procedures can lead to inaccurate use of nutrients and late detection of diseases. Also, as farm sizes increase and the availability of labor decreases, more effective agricultural practices are necessary. A high frequency, high resolution, and optimally planned crop monitoring apparatus, collaboratively supervised by a human operator to reduce cost, reinforced by agile robotics and spectral sensing technologies could lead to intelligent, efficient, safe, and more effective biotic and abiotic stress management.

The research of this thesis is part of bigger research, which its objective is to develop and enable, for the first time, a human integrated intelligent sensory-robotic system for inspection and early detection of biotic and abiotic stresses and risks in greenhouse crops. This novel system will reduce human labor, reduce the amount of misused watering, pesticides and fertilizers, and detect and prevent, in time, the spreading of diseases.

This thesis will try to develop a method to find an optimal manipulator kinematic design to detect stresses in greenhouse crops, using evolutionary algorithms and set-based concept approach.

This thesis focusses on a preliminary evolutionary search process. The aim of such a preliminary process is a reduction of the amount of the considered concepts into a manageable subset that includes superior concepts. In this thesis, superior concepts are concepts that satisficing Window of Interest that is updated dynamically during the running of the algorithm.

2. Literature Review

2.1. Robotics

2.1.1. Robot

International Organization for Standardization (ISO) defines robots and robotic devices in [ISO 8373](#). ISO 8373 defines robot as: “actuated mechanism programmable in two or more axes with a degree of autonomy, moving within its environment, to perform intended tasks.”

The standard is also classified robot into an industrial robot or service robot.

While the industrial robot is a manipulator with more than 3 DOF which can be either fixed in a place neither mobile, the service robot is robot that performs useful tasks for humans or equipment excluding industrial automation applications.

Robots are widely used, most of the robots today are industrial robots used in automotive factories, medical field, construction, space, agriculture, research and more. Another type of robot that is widely used is domestic robots, type of service robots, such as robotic vacuum cleaners, for kitchen, lawnmower and more.

2.1.2. Robotics in Agriculture

Robotics in agriculture is mainly used in seed mapping, weed mapping, and micro spraying[1], but also includes transplanting, irrigation, harvesting and plant protection in the field, greenhouse and aquaculture as well.

The main issue for agricultural robots is the ability to operate in unstructured agricultural environments with the same quality of work achieved by current methods and means[2].

The unstructured environment is expressed in variation among crops (tomato, cucumber, pepper), the variety of objects in a crop (crop vary in position, size, shape, etc) and variation in the environment (orchard, field, greenhouse)[3].

Another major issue is the costs of robotic systems must be sufficiently low to economically justify their use since the agricultural produce being dealt with is of relatively low value. However, the recent cost reductions in electronics, computers, and robotics should enable such systems to penetrate more widely into agriculture[2].

The robot tasks and operations in agricultural are[4]:

- Weed control and disease monitoring: A major challenge in agriculture is the inability to detect stresses and risks early enough preventing the uncontrolled spreading of stresses causing irreparable damage, which causes waste of pesticides and fertilizers.
- Navigation and guidance: Navigation and guidance are the basic parts of automation for agriculture. They include three levels of autonomy: conventional

steering, an operator-assisted or automatic system (supervised by a human operator), and a fully autonomous system.

- Transplanting and seedling: Automating transplantation is feasible in several situations, mainly when the operations are repeated every few weeks in the same plot, as is the case with leafy vegetables or herbs.
- Pruning and thinning: Pruning is a labor-intensive task used to control tree shape, increase exposure to sunlight and remove protruding branches.
- Harvesting: Harvesting/fruit-picking is one of the most common tasks in agriculture and also one of the most demanding and challenging areas for agricultural robotics.

2.1.3. Robotics Manipulators

2.1.3.1. *Serial Manipulator Kinematics*

A serial manipulator consists of a fixed base, a series of links connected by joints, and ending at a free end carrying the tool or the end-effector[5].

The study of Kinematics is essential to Robotics. A robot, to perform most applications, needs to process positional data and transform data from one frame of reference to another[6]. In the thesis, the kinematics format used is Unified Robot Description Format (URDF). URDF is an XML(extensible Markup Language) format for representing a robot model. It represents the appearance of the robot and its intended action[7]. Degrees Of Freedom (DOF) is defined by Chebychev–Grübler–Kutzbach criterion[5] for motion in 3D

$$DOF = 6(N - J - 1) + \sum_{i=1}^J F_i \quad (2.1)$$

Where N- number of links, J number of joints that connect 2 links, F_i joint's freedom of the i^{th} joint.

In serials manipulators

$$N = J + 1 \quad (2.2)$$

Therefore

$$DOF = \sum_{i=1}^J F_i \quad (2.3)$$

2.1.3.2. *Robotics performance indices*

One way to classified performance indices is based on their scope (local \ global).

Local indices are performance metrics that are dependent on the posture of the manipulator and are also known as posture-dependent indices[8].

Global indices are posture independent indices. They represent a global characteristic of the manipulator's workspace. Global indices are needed to compare the structure and behavior of two manipulators that perform the same task[8].

Manipulability index is a local index, which based on the Jacobian matrix is a kinematic performance index[9]. The manipulability index can be calculated in two forms:

The first form is

$$\mu = \sqrt{|J * J^T|} \quad (2.4)$$

where J is the Jacobian in the current position.

The second form is

$$\mu = \sigma_1 * \sigma_2 * \dots * \sigma_m \quad (2.5)$$

Where m – number DOF, σ_i – the i^{th} singular value of the Jacobian.

The value of the Jacobian varies between [0-1] when 0 means the manipulator in singularity and 1 that the manipulator is far from the singularity.

Condition Number is also a local and kinematic performance index based on the Jacobian matrix[10]. Condition number measures the independencies of the columns of the Jacobian[8]. As the manipulability index, calculated as follow:

$$k = ||J|| * ||J^{-1}|| \quad (2.6)$$

Where $||J||$ is the norm of the Jacobian matrix.

Condition number doesn't have an upper bound $k \in [1, \infty]$ so to avoid computational problems Local Condition Index is used.

Local Condition Index (LCI), which is more commonly used, is calculated as follow:

$$LCI = \frac{1}{k} \quad (2.7)$$

Where k is the condition number.

$LCI \in [0, 1]$ where 1 the manipulator is in isotropic points: the Jacobian matrix columns are orthogonal and its column vectors are of equal magnitude.

Joint Mid-Range Proximity Index is defined for the avoidance of joint limits and to maintain the joint displacement as close to mid-range as possible[8]. Baron[11] defined it as follows:

$$z = \frac{1}{2} * (\theta - \bar{\theta})^T * W * (\theta - \bar{\theta}) \quad (2.8)$$

Where W is a positive-definite weighing matrix, θ is the current joints position and

$$\bar{\theta} = \frac{1}{2}(\theta_{max} + \theta_{min}) \quad (2.9)$$

Degree of Redundancy The degree of redundancy (r) is equal to the number of degrees of manipulator freedom (n) less the rank of the workspace (m), given as[12]:

$$r = n - m \quad (2.10)$$

Redundancy is the ability of a manipulator to reconfigure itself with the end-effector remaining in a fixed position[8].

2.2. Optimization

Successful engineering design generally requires the resolution of various conflicting design objectives. One of the most powerful tools for resolving such objectives is multiobjective optimization[13].

2.2.1. Evolutionary Algorithm

Genetic Algorithms (GAs) and Evolution Strategies (ESs) are both Evolutionary Algorithms [EA], an optimization search algorithm and based on the principles of the survival of the fittest and natural selection, which are the laws of natural evolution argued by Darwin[14].

EA relies upon the collective learning paradigm gleaned from natural evolution and implement the principles population, mutation, crossover, and selection. Besides, ESs try to use a collective self-learning mechanism to adapt its strategy parameters during the optimum search (adaptive search)[15].

Basic EAs are characterized by the genetic operations of selection, crossover, and mutation [16].

The first step in the algorithm, after initializing the population, is fitness. During each generation, the population is evaluated for fitness according to the ability to attain a satisfactory result[16].

The next step is the stop condition if the EA fulfills one of the criteria to stop, for example, how many generations to run, running time, etc.

If the EA didn't stop in the stop condition, it will get into the selection step. In this step, the best population, according to fitness, has the most likely to continue to the next step. There are many types of selection methods, presented here two basics selection methods:

- Tournament selection: involves running several tournaments among a few randomly selected individuals from the population, and the individuals with the best fitness continue
- Roulette Wheel Selection(RWS): The fitness function assigns fitness to possible solutions. This level of fitness is used to link a probability of choice to everyone.

The next step is genetic operators to create the next generation of the population. The genetic operators are crossover and mutation.

The role of crossover is to generate better solutions by exchanging information contained in the current good solution.

Because of the low disruptiveness of the crossover, disruptive mutation operators are needed for the EA to continue to search without premature loss of genetic diversity[17].

While in GA the crossover is the main genetic operator in ES the mutation is the main operator and crossover needs to be used in cases where each child has multiple parents.

ES usually include elitism. Elitism guarantees that the solution in the next generation won't decrease from the previous generation.

GA more likely to find the global maximum and usually slower and used when an optimal solution is needed and more common for solving computational problems.

ES is usually faster but find a local maximum, so it good when a good enough solution acceptable. ES more common to solve engineering problems.

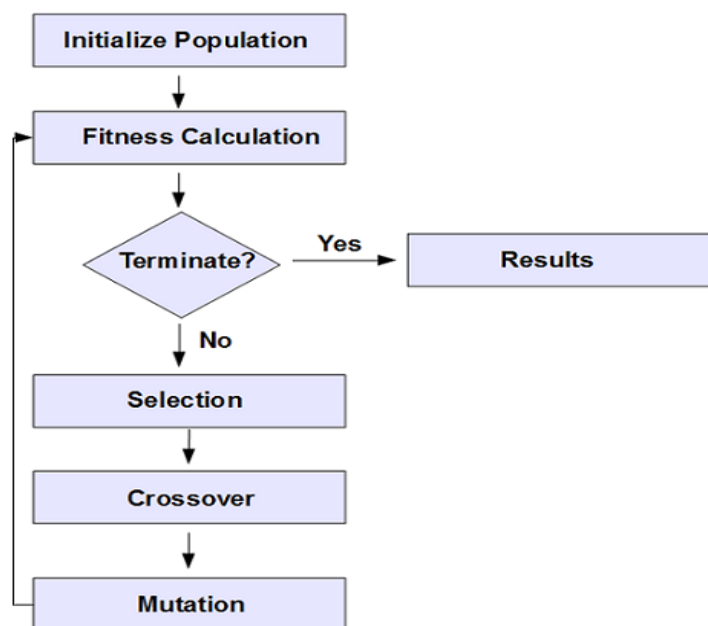


Figure 1 - Basic EA

2.2.2. Pareto Front

By definition, a search and optimization problem with multiple conflicting objectives resorts to a set of optimal solutions known as the Pareto-optimal solutions[18].

Pareto is a state of allocation of resources from which it is not possible to reallocate to benefit one individual or one preference without adversely altering the condition of at least one individual or preference.

All the individuals in the Pareto front represent non-dominated values[19].

Solution x is said to be dominate about solution y if solution x is no worse than solution y in all objects and solution x is better than solution y at least in one objective[20].

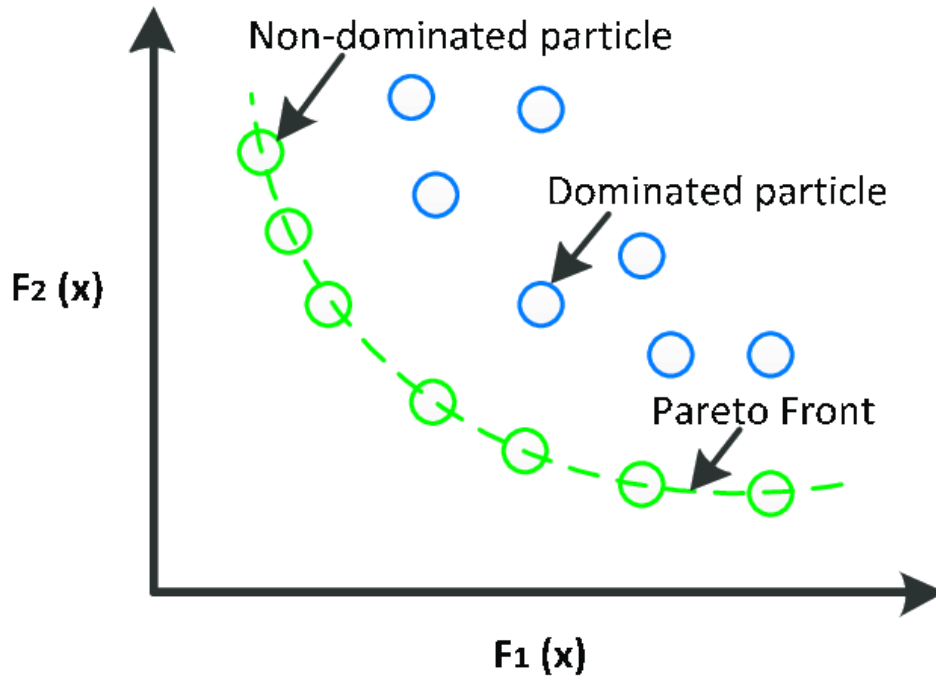


Figure 2 - Pareto Front & Dominance[21]

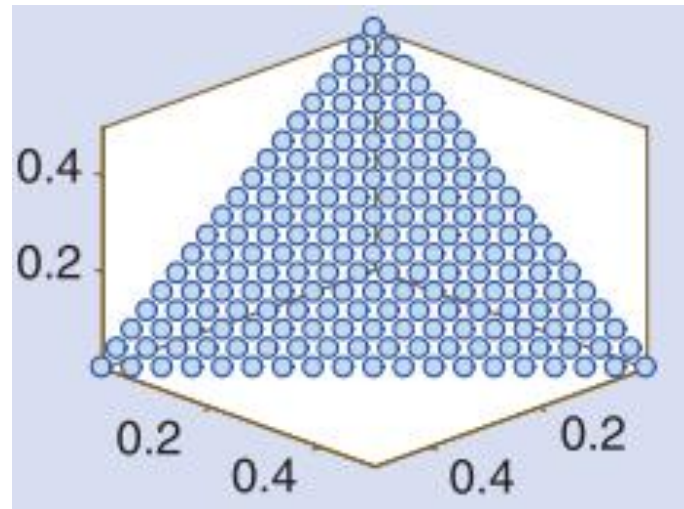


Figure 3 - Pareto Front DTLZ1[22]

2.2.3. Window Of Interest

The Window Of Interest (WOI) indicates what is considered as an acceptable performance vector. Rather than being interested in finding concepts' fronts, here the designers are interested in finding which of the considered concepts have at least one solution with a performance vector within a pre-defined WOI. Concepts that meet this requirement are considered satisficing[23].

In decision making, satisficing refers to the use of aspiration levels when choosing from different paths of action. By this account, decision-makers select the first option that

meets a given need or select the option that seems to address most needs rather than the "optimal" solution.

The two main advantages of WOI compared to concept based Pareto method are first, in the WOI, there is no need to find any front, second, the WOI, within which solutions are sought, is pre-defined[23].

2.2.4. set-based concept

The set-based concept approach has been suggested as a means to simultaneously explore different design concepts, which are meaningful sub-sets of the entire set of solutions[23].

It involves a comparison of subsets of the solution set, where each such sub-set represents a conceptual design solution. The use of a set to represent a concept reflects the fact that a design concept is not a final solution[24].

set-based concept search approach is not optimization but gaining general knowledge of the design space. The approach to design space exploration includes predefined design concepts that are used to explore the design space at both the level of concepts and the particular designs that accompany it. This is achieved by the set-based concept approach[25].

2.3. Manipulators Optimization

In the past, the mechanical design of robotic manipulators has been based on some simple criteria, such as the number of joints, link sizes and weights, and payload capacity. The design process includes deciding upon a robot configuration largely by intuition or experience[26]. Those days research efforts have been divided between those concerning performance measures (indices) and those that deal with optimization approaches and techniques(Ami draft paper).

Some of the optimizations studies focused on optimizing the kinematic design such as geometric parameters, joint travel ranges, and tolerances, links lengths, manipulator workspace[26].

Other studies focused on optimizing the dynamic design such as the link masses, torques, inertias, and material properties and some studies tried to optimize by kinematics and dynamics.

Most studies on optimizing manipulators employ numerical methods. Early attempts to numerically optimize manipulators employed gradient-based methods.

Nevertheless, using gradient-based methods for optimizing manipulators is very difficult[27]. GA have become popular since the ninetens for solving such optimization problems.

3. Methods and materials

3.1. Manipulator arms

3.1.1. Definition

Each manipulator arm configuration () consists of three features: joint, joint axis and link length. The joints are divided into three types:

- Roll – Revolute joint about Z-axis
- Pitch – Revolute joint about Y-axis
- Prismatic – Linear sliding along Z-axis

The Joint axis is defined relative to the previous coordinate system, therefore according to the previous joint axis.

The link length is the length of each link along the Z-axis of its coordinate system.

Figure 2 gives an example for Roll_Z_0.1 → Pitch_Y_0.4 → Prismatic_Y_0.7.

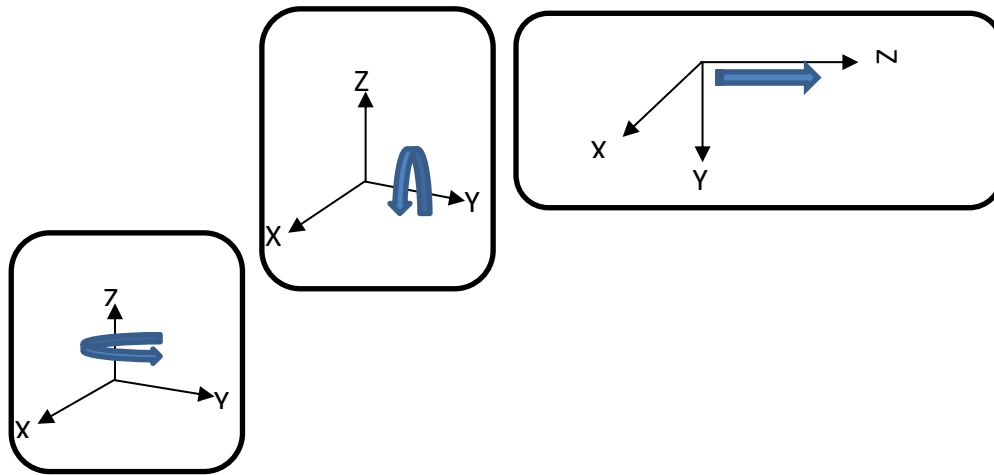


Figure 4 - Configuration Example

3.1.2. Parameters

For the construction of each configuration the following parameters have been set:

- All the links are cylindrical where at radius of 0.05m, 0.045m, 0.04m, 0.035m, 0.03m, 0.025m for 1,2,3,4,5,6 DOF, respectively.
- For Revolute joints (Roll and Pitch) the joint range is $[-\pi, \pi]$ and for prismatic joints, the range is $[0, 2 \times \text{link length}]$
- All the configuration will be between 3-6 DOF
- The length of the links will be 0.1, 0.4, or 0.7 meters
- Success: the arm need to reach the middle and the lower points and one of the top points (Figure 3)

This gives about 34,636,800 theoretic configurations.

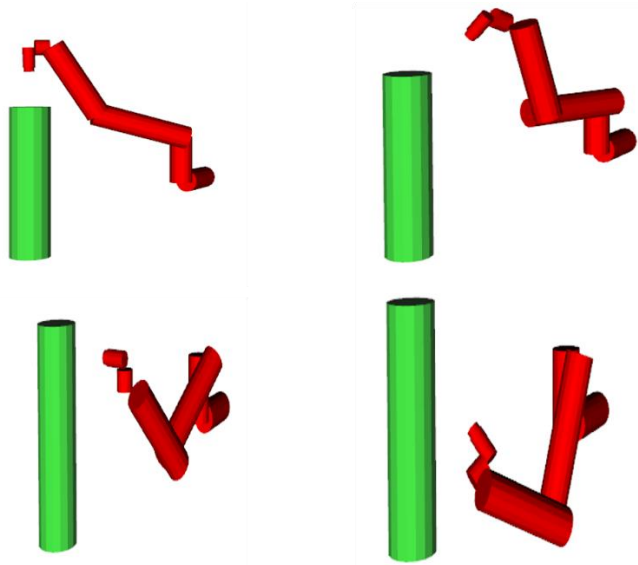


Figure 5 - Manipulator detection points

3.1.3. Assumptions

To reduce the number of configurations and to fit the specific problem some assumptions have been made:

- The first joint is rotational along Z ax (Roll_Z)
- The first link length = 0.1m
- The total length of all the links > 1m
- 2 adjacent prismatic joints must be perpendiculars
- No more than 3 prismatic joints
- After a Roll joint, the next joint will not be Roll\Pitch joint in the Z-axis
- After a Roll joint, the next joint will not be in the X-axis
- After a Pris Z joint that its previous joint is a Roll joint, the next joint will not be in the X-axis

those assumptions reduced the available manipulators to 1,695,044 configurations.

3.1.4. Indices of manipulator performance

To make comparisons between the configurations and to be set as the optimization objectives several manipulators Indices and other indices have been checked:

All the indices that been checked are local, meaning indices that only depend on a specific position of the arm or the arm structure. Because the indices are calculated for several points the point with the lowest grade was selected as the grade of the configuration.

The chosen indices from the literature presented in **Table 1**.

Since the upper limit of the Mid-Range Proximity Index is different for each configuration and to regard the differences between the type of joints, the value of the joint current position was normalized by the joint range, which bounds the index between [0 – 0.5].

Table 1 - Selected Indices

index	expression	explain	min value	max value	notes
Manipulability index	$\mu = \sqrt{ J * J^T }$	J is the jacobian	0	1	The Yoshikawa manipulability index happens to be the most widely accepted and used measure for kinematic manipulability
Mid-Range Proximity	$z = 0.5 * (\theta - \bar{\theta})^T * m$ $m = \begin{bmatrix} \theta_1 - \bar{\theta}_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \theta_n - \bar{\theta}_n \end{bmatrix}$	$\bar{\theta} = 0.5 * (\theta_{max} + \theta_{min})$ $\theta_{max} \& \theta_{min} - joint\ limits$	0	0.5	take into consideration the fact that the joint's motion is limited to various degrees.
Degree of Redundancy	$r=n-m$	n: manipulator DOF [3-6] +2 m: workspace DOF [6]	-1	2	Redundancy is the ability of a manipulator to reconfigure itself with the end-effector remaining in a fixed position
			the range in our case		

3.2. Simulation

3.2.1. Simulator

Gazebo simulator and MoveIt have been used for motion planning, inverse kinematics, control, and collision checking, to simulate different configurations of robotic arms. Robot Operating System (ROS) has been used to connect between those platforms and simplify the process.

Gazebo is a simulator tool that simulates the real world and the environment where the selected manipulator arm will work. In the simulation, the plant is modeled as a cylinder at a height of 0.75 meters and a radius of 0.5 meters (green cylinder in figure 3).

MoveIt is a tool to calculate motion planning and control the movement of each configuration in the Gazebo simulator. the motion planning algorithm that selected for each configuration is rapidly exploring random tree (RRT) algorithm, which can handle problems with obstacles and differential constraints. RRT isn't finding the optimal path but find a valid path in a short time. In the simulation, the simulation running time is very important and the RRT has been limited to 2 seconds to find a possible path.

The dynamic and kinematic structure that is used in ROS and Gazebo, to present arms, is Universal Robotics Description Format (URDF). Because the URDF also considers the dynamic part of the arm there is necessary to calculate the weight of each link. The

weight of each link is calculated by the ratio between the accumulated length to the accumulated weight of the link- the ratio was calculated according to 2 different types of manipulators: UR5 and MOTOMAN YR-MH005LN (figure 4)

In the simulator, 4 points have been chosen to be reached by the arm. All the points are at a distance of 30 cm from the plant and in a different orientation.

Since the manipulator will be mounted on a mobile platform that will drive through the greenhouse, a prismatic joint, parallel to the plant for each configuration, was added in the base of the arm. The range of this prismatic joint is 1.5 meters so it will be able to move before in after the plant and the mobile platform position won't be the reason that the configuration didn't reach the desired point.

On the arm end effector, a multi-spectral camera will be mounted to conduct early detection of biotic and abiotic stresses in specialty crops, for this purpose, the image orientation is not important, therefore, a rotation joint on Z-axis was added to the end of each arm.



Figure 6 – Motoman(left) & right UR5(right)

3.3. Optimization

3.3.1. the set-based concept approach

As mentioned earlier there are 1,701,647 configurations. with an average simulation run time of 15 seconds per configuration, examining all configuration will take about 300 days on a Virtual Machine with the following specification:

Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz, 8GB RAM and 90GB of storage.

Therefore, an optimization method was developed to find the optimal configuration in a reasonable time. In this case, the set-based concept approach, which combined configurations with the same properties in one design concept. In this specific problem, there are almost 800 different concepts and it will be time-consuming to make a deep search of each concept. therefore, a multi-objective evolutionary search for satisficing concepts based on dynamic window-of-interest was developed. the aim is to reveal which of the concepts have at least one solution with a performance vector within a dynamically changed window-of-interest. In the first part, a predefined

computing time allocated to find satisficing concepts that would be explored more deeply in the second part of the optimization.

In the second part, the concepts that satisfied in the previous part are going to be explored more widely and to find the Pareto front of the problem by using a multi-objective evolutionary algorithm.

3.3.2. Objectives

The Objectives of the optimization problem selected from the indices of the previous section and are the following:

- Maximum manipulability [0-1] (f1)
- Minimum Z (Mid-Range Proximity) [0 -0.5] (f2)
- Minimum degrees of Freedom [4 - 6] (f3)

This optimization problem is mixed with minimum and maximum objectives, to change the problem to be minimum problem f1 objective has been changed as follow:

Min (1 – manipulability) [0-1] (f1)

3.3.3. Independent variables:

The independent variables of the optimization problem are:

- X1: Joints Types: array [Roll, Pitch, Prismatic]
- X2: Previous ax: array [X, Y, Z]
- X3: Links Lengths: array [0.1 ,0.4, 0.7] (meters)
- X4: Number Degrees of Freedom: Int [3, 4, 5, 6]

3.3.4. Constrains

The constraints of the optimization problem are derived from the assumptions of the simulator section and are as follows

1. The first joint is rotational along with Z-axis:

$$X1[0] = \text{Roll}, X2[0] = Z$$

2. First link length = 0.1m :

$$X3[0]=0.1$$

3. The total length of all the links > 1m :

$$\text{Sum}(X3) > 1$$

4. 2 adjacent prismatic joints must be perpendiculars:

$$\text{If } X1[i]==\text{Prismatic } X1[i]==\text{Prismatic} \text{ then } X2[i] \neq Z$$

5. No more than 3 prismatic joints

6. After Roll joint can't be Roll\Pitch joint in the Z-axis:

$$\text{If } X1[i]==\text{Roll} \text{ and } (X1[i+1]==\text{Roll} \text{ or } X1[i+1]==\text{Pitch}) \text{ then } X2[i+1] \neq Z$$

7. After a Roll joint, the next joint will not be in the X-axis

if (X1[i]==Roll) than X2[i+1]!=X

8. After a Pris Z joint that its previous joint is a Roll joint, the next joint will not be in the X-axis

if ((X1[i-1]==Roll and X1[i]== Pris and X2[i] == Z)) than X2[i+1]!=X

9. Joints limits:

Roll\ Pitch [0-360°] and Pris [0 – 2*link length]

10. Number of detection points: 4

11. Success: the arm need to reach to one of the 2 top points and the middle and the lower points and one of the top points

3.3.5. Concepts

Each concept is defined from the following variables:

- 1) DOF – Number degrees of freedom of the configuration
- 2) Pitch joints – Number of pitch joint in the configuration
- 3) Long Link - Number of long links (0.7m) in the configuration
- 4) Acc length – Accumulated length of all the links of the configuration. This variable is defined as a range of distances
- 5) # of Parallel axes Y – Number of parallel axes along world Y ax in a row
- 6) Longest Link – What is the longest link in the configuration
- 7) P/R ratio – The ratio between Prismatic joints to Revolute joints (Roll\Pitch)

All the variables and their possible values are presented in **Table 2**.

Not all the combinations between the variables are possible. For example, it's not possible to concept to be with 4 DOF and 5 Pitch joint.

The total amount of concepts that received is 794. The number of configurations in each concept varies from 1 to 68520.

Table 2 - Concepts

	Variable	number of options	1	2	3	4	5	6	7	8
1	DOF	3	4	5	6	***	***	***	***	***
2	Pitch joints	6	0	1	2	3	4	5	***	***
3	Long links (0.7 m)	6	0	1	2	3	4	5	***	***
4	acc length	5	1-1.5	1.6-2	2.1-2.6	2.7-3.1	3.1-3.6	***	***	***
5	# of parallel axes – y	5	0	2	3	4	5	***	***	***
6	longest link	2	0.4	0.7	***	***	***	***	***	***
7	prismatic to revolute ratio	8	0	0.2	0.25	0.33	0.67	1	1.5	3
	Total	794								

3.4. An interface between the simulator and optimization algorithms

In order the simulator will be able to communicate with an optimization algorithm, an interface was built in Python. The interface is built from two parts, the configuration builder and indices calculator. In **Figure 5** it can be seen a scheme of the interaction between the interface and the simulator and the optimization method.

3.4.1. Configuration builder

The configuration builder gets from the optimization algorithm the selected variables by the algorithm to be simulated (DOF, Joints Types, Joints axes, Links Lengths) and creates from this data a URDF file which contains the kinematic and dynamic representation of the manipulator. After creating the URDF file the interface enters it with the predefined detection points into the simulator.

3.4.2. Indices Calculator

The indices calculator gets in return, from the simulator, if the configuration succeeded to reach the desired points and if it's succeeded what was the Jacobian and joints position at every point. The indices calculator uses the Jacobian to calculate the manipulability index and the joints position to calculate the mid proximity joint index. After calculating the manipulability index and the mid proximity joint index, the indices calculator returns those indices to the optimization algorithm to evaluate this configuration.

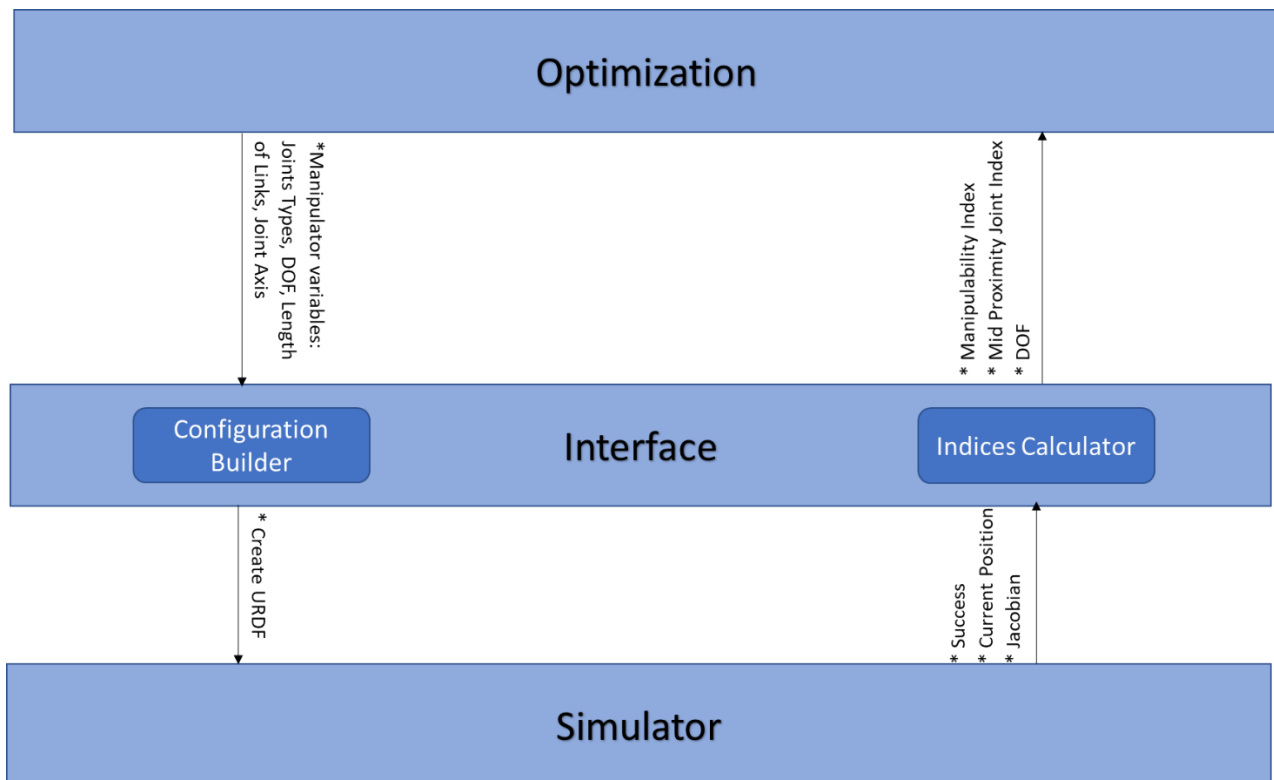


Figure 7 - Interface between optimization to the simulator

3.5. Preliminary Window of Interest

As described before, the optimization is conducted in 2 steps: the first step is to find several concepts from the 794 concepts that satisfying The concept that satisfying is a concept that is at the Window Of Interest (WOI). To start this step in reasonable WOI, random configurations were simulated.

3.5.1. Configuration selection

The configurations were selected from all the concepts of inconceivable selection.

The selection is done as follows:

- Simulation of 60000 selected configurations - One week of computer time (there were 1.5 available computers) – each configuration takes about 15 seconds to simulate
- From concepts with less than 25 configurations, all the configurations selected
- From concepts with more than 25 configurations, selected 25 configurations or 3% of the configuration, what higher
- All the selected configurations were tested in the simulator

3.5.2. Pareto Front

After all the selected configurations were simulated a Pareto front is calculated.

The Pareto Front was calculated as follow:

- All the configurations are plotted in a 3D space according to their results when the axes are the 3 objectives: Manipulability Index, Mid-Range Proximity Index, and DOF.
- All the 'Non-Dominated' configurations building the Pareto front

The calculated Pareto front can be built from several concepts, mustn't be from one concept, and will be set as the WOI of the next part.

3.6. Evolutionary optimization with Dynamic-Window of interest

In this type of algorithms, the WOI is dynamic (DWOI), meanings that WOI updated during the processes and continues to approach the origin. The way of calculation the initial WOI described in the previous section. The mating is done only inside each concept, isn't done between concepts. Concepts with a small number of configurations the selection will be randomly and concepts with a large number of configurations the selection will be done by the genetic algorithm.

3.6.1. Evolution Strategies

In this part, Evolution Strategies (ES) is used to find the concepts that satisfying the DWOI and will be good enough to continue to the next part that will find the optimal configuration from the satisfying concepts.

The ES runs only inside each concept and has no effect on other concepts except for changing the DWOI.

The ES that used in this case is designed as following (figure 6):

To speed up the ES, all the concepts with up to 1000 configurations were simulated before, this action takes 16 days of computation time and all the concepts with 5dof simulated before as well, which takes 5 more days of computation and helps to handle the resources.

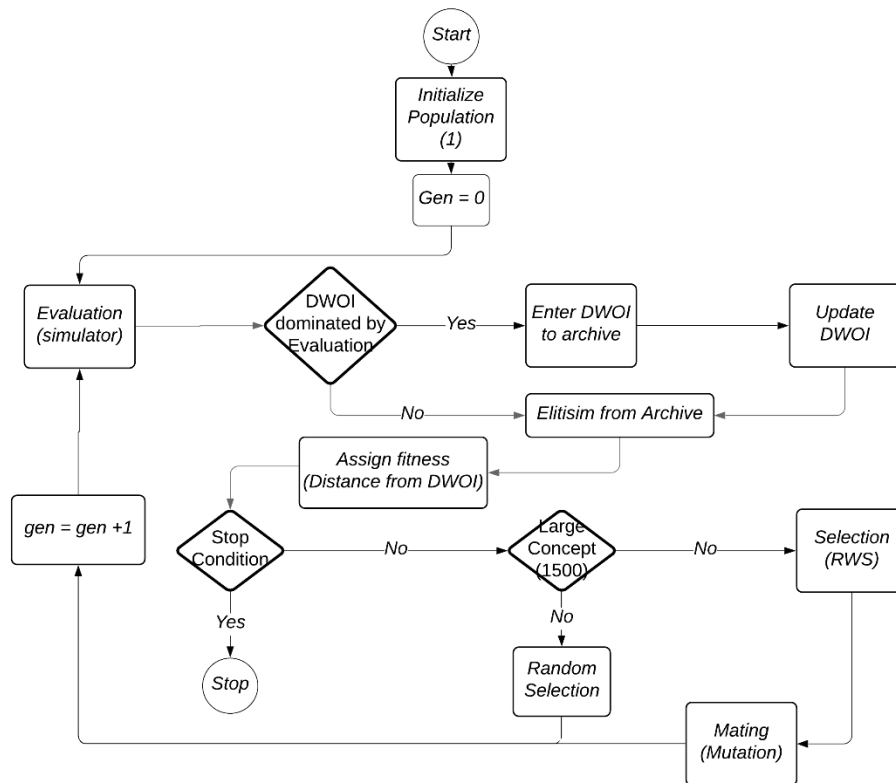


Figure 8 - Evolution Strategies

3.6.1.1. Initializing & Evaluation

Each concept starts with a random population of one. The evaluation is done by simulation of the selected configurations and calculation of their indices.

3.6.1.2. DWOI

The calculated indices are checked, and if one of the configurations is dominating one of the configurations in the DWOI it replaces it. After the update of the DWOI, the old DWOI is entered into the archive. After the domination check and DWOI update (if needed) a fitness will be assigned to each configuration.

3.6.1.3. Elitism

Before Assign Fitness elitism is performed. The elitism is done by using an archive with previous Non-dominated results and it guaranty that only parents with good genes will enter into the mating pool.

3.6.1.4. Assign Fitness

The fitness is assigned by calculating Euclidean distance for each configuration from the DWOI when the configurations with the smallest distance are getting the higher fitness. The fitness is calculated as follow:

$$fi = e^{-3*d}$$

When fi is the fitness of the i -th element and d is the distance from the DWOI.

3.6.1.5. Stop Condition

After the assign fitness step, there is a stop condition step. In this step, it's checked if one of the following conditions are fulfilled:

1. All the configurations in the concepts are evaluated
2. The mating doesn't succeed to generate more offsprings
3. All the remaining concepts are in DWOI or stopped
4. Arrived at generation = 1240
5. If the predefined time of 7 days is passed.

Conditions 1-2 are local stop conditions, it means that they are stopping only the concept and not all the process as conditions 3-5.

If the stop conditions aren't fulfilled the algorithm continues for creating new configurations.

~~With 7 days of optimization run, 15 seconds per configuration the possible options of configurations to simulate are 40320 configurations.~~
~~with the fastest possible resource allocation (detailed in resource allocation section 3.6.2) and~~

3.6.1.6. Selection

If the concept is small (a concept with less than 1500 configurations) than the new configurations are selected randomly from the concept's configurations. But if the concept is large, then the configurations' fitnesses are entered into the selection, which performed by Roulette Wheel Selection (RWS).

3.6.1.7. Mating

The mating step builds only from mutation.

Each configuration disassembled to parts in the same number as it DOF.

Each part of the configuration contains Joint, Axes and Link length. For example, Roll z 0.1.

As shown in Table 3, each part gets a suited number between 1-29 (except 10 & 20), and this number will be its id number.

After each part gets its id, two types of neighbors were calculated (Table 4). The 1st type is first-order neighbors when changing only one feature in the part (axes or link length) gives us another part which doesn't give configuration outside the concept, as if the Joint type changed or link length is 0.7 do. The 2nd type is second-order neighbors when two features are changed to get part from another part.

For example, id 1- Roll z 0.1 the first-order neighbors are id 2- Roll z 0.4, id4- Roll y 0.1 and id7 – Roll x 0.1. The second order for id1- are id 5- Roll y 0.4 and id 8- Roll x 0.4.

The mutation is performed as a following:

- 1) Select Random Number between 1-5 (arm index)
- 2) Select Random neighbor from the first neighbors
- 3) In the selected arm index replace the link with the selected neighbor
- 4) Check the new configuration is inside the concept and not simulated before
- 5) If 4 is true continue
- 6) Else return to 1

* Step 2 includes also the second neighbors for all the concepts after 100 generations or before to specific concept if the concept $C_r = 0$.

Table 3 - Parts ID

1 Roll Z 0.1	11 Pitch Z 0.1	21 Pris Z 0.1
2 Roll Z 0.4	12 Pitch Z 0.4	22 Pris Z 0.4
3 Roll Z 0.7	13 Pitch Z 0.7	23 Pris Z 0.7
4 Roll y 0.1	14 Pitch y 0.1	24 Pris y 0.1
5 Roll y 0.4	15 Pitch y 0.4	25 Pris y 0.4
6 Roll y 0.7	16 Pitch y 0.7	26 Pris y 0.7
7 Roll x 0.1	17 Pitch x 0.1	27 Pris x 0.1
8 Roll x 0.4	18 Pitch x 0.4	28 Pris x 0.4
9 Roll x 0.7	19 Pitch x 0.7	29 Pris x 0.7

Table 4 - ID and its Neighbors

Neighbors			Neighbors			Neighbors		
Cell	1st	2nd	Cell	1st	2nd	Cell	1st	2nd
1	2,4	5,8	11	12,14,17	15,18	21	22,24,27	25,28
2	1,5,8	4,7	12	11,15,18	14,17	22	21,25,28	24,27
3	6,9		13	16,19		23	26,29	
4	5,1,7	2,8	14	15,11,17	12,18	24	25,21,27	22,28
5	4,2,8	1,7	15	14,12,18	11,17	25	24,22,28	21,27
6	3,9		16	13,19		26	23,29	
7	8,1,4	2,5	17	18,11,14	12,15	27	28,21,24	22,25
8	7,2,5	1,4	18	17,12,15	11,14	28	27,22,25	21,24
9	3,6		19	13,16		29	23,26	

3.6.2. Resource Allocation

To get the best results 2 types of resource allocation heuristics been tested: Fair and Greedy. The two heuristics were tested on the same ES, given the same time of computation and run on the same hardware, to prevent bias to one of the heuristics. In both heuristics concepts that **inside** the DWOI will not get resources as long as they **inside** the DWOI.

3.6.2.1. Fair Heuristic

Each concept gets the same resources regardless of its population size and how it progresses. The rationale behind this heuristic is all the concepts need to be treated in the same way.

3.6.2.2. Greedy Heuristic

In this heuristic, the resources are allocated during the running of the ES according to the convergence rate of the concepts. Concepts with a high convergence rate will get more resources than concepts with a low convergence rate.

Concept Convergence Rate($C_{r,i}$) - for i-th concept its distance from the origin of the axis in generation 0 (d_0) and in generation X (d_X) calculated and their difference will be divided by the number of generations X.

Convergence Rate of the i-th concept $C_{r,i} = \frac{d_{X,i} - d_{0,i}}{X}$ where $i=1, \dots$, Number of concepts.

The distance from the origin is calculated as follows: from the i-th concept all the Non-dominated results are taken and the distance from each point is calculated from the origin. The minimum distance is the concept distance from the origin.

Greedy heuristic algorithm:

- 1) In the first 10(**% or generations**) all the concepts get the same resources

- 2) After the first 10(% or generations) Concept Convergence Rate are calculated for each concept - $C_{r,i}$
- 3) Assigning $C_{r,i}$ in 3 sets: when each set is sorted descending order
 - set 1: $C_{r,i} > T_{high}$
 - set 2: $T_{high} > C_{r,i} > T_{low}$
 - set 3: $C_{r,i} < T_{low}$
- 4) The best 90% of the concepts will continue as follow: all the concepts in set 1 (even if they are more than 90% of the concepts). If all the concepts in set 1 are less than 90% of the concepts than the remaining concepts will come from set 2 and if still not enough from set 3.
- 5) The concepts that continue will get the same resources
- 6) The concepts that not continue their $C_{r,i}$ will be saved and in the next resource allocation step can get resource again if their $C_{r,i}$ will be better than another concept, if they are belonging to set 2. If they are belonging to set 3 they will be eliminated and won't get more resources.
- 7) If none of the globals stop conditions occurred go back to step 2

Where $T_{high} = 0.01$ (high threshold) and $T_{low} = 0.05$ (low threshold)
 in step 4 there is a problem of getting a very small number or even 0 concepts,
 therefore the minimum number of concepts is 10% of the total number of concepts.

3.7. Find Pareto front using a multi-objective evolutionary algorithm

In this part, the concepts selected in the previous part entered into an Evolutionary Algorithm for multi-object optimization. The algorithm selected is ***** because of *****.

- * Select algorithm
- * Compute Indices

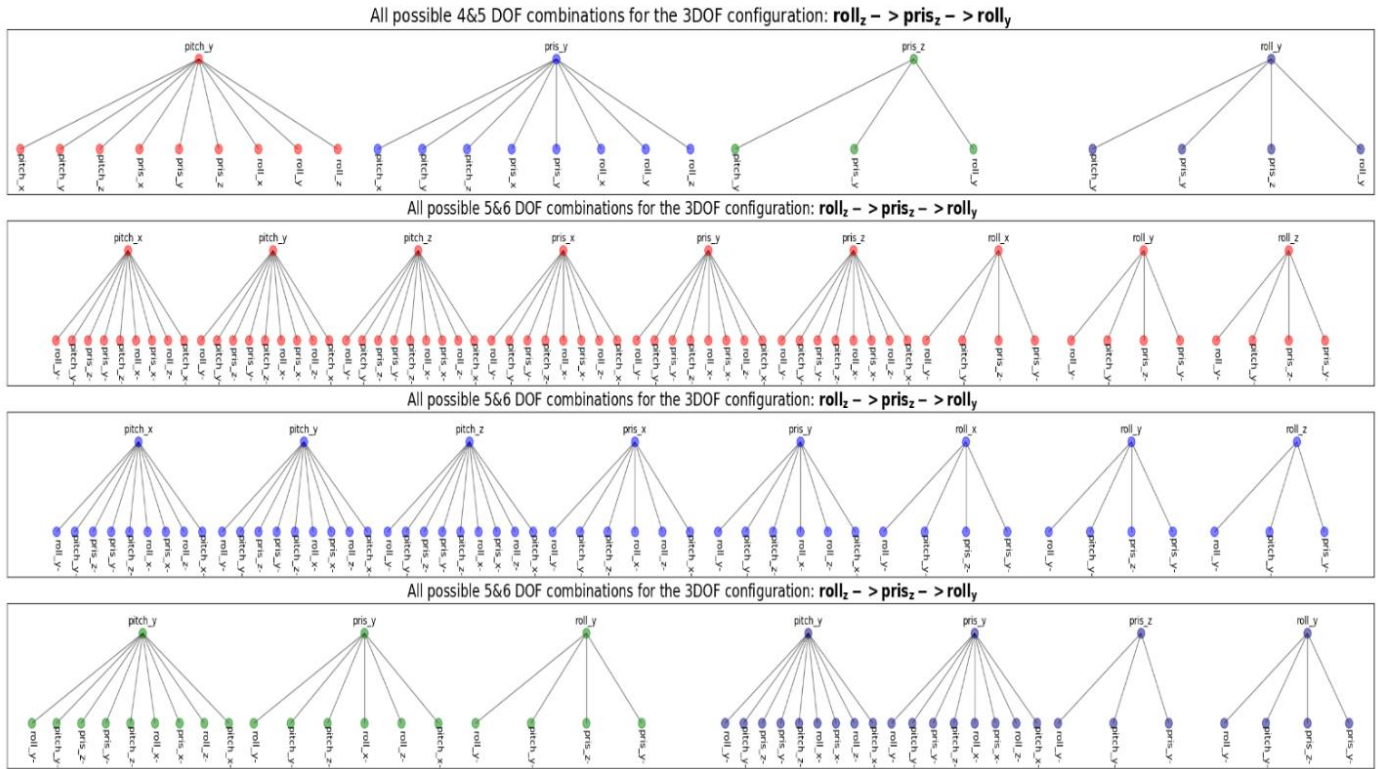


Figure 10 - All Joints & Axes possibilities one branch

4.1.2. Indices of manipulator performance

For choosing the indices to be used a preliminary simulation conducted over random 127000 configurations, all 3&4 DOF and part of 5&6 DOF.

From the preliminary simulation, none of the 3 DOF configurations succeeded to reach to desired points, therefore 3 DOF will not check further.

The tested indices of manipulator performance were :

- Manipulability Index
- Local Condition Number (LCI)
- Degrees of freedom (DOF)
- Time
- Mid-Proximity Joint Index

In **figure 9** it can be seen that there is a linear relationship between LCI and Manipulability, which makes sense because both of them are checking how close the manipulator to singularity.

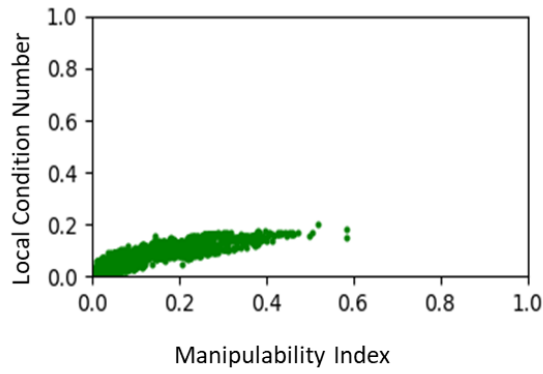


Figure 11 - LCI Vs Manipulability Index

In figure 10 it can be seen that no connection can be found between the Manipulability Index and the Mid-Proximity Joint Index.

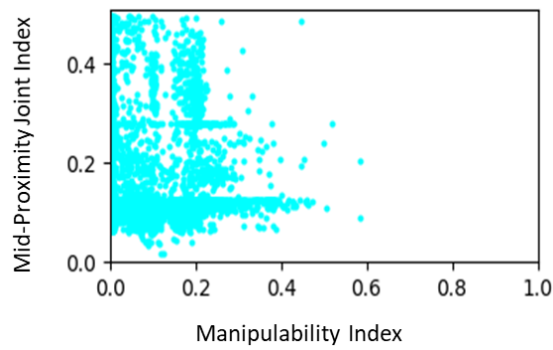


Figure 12 - Mid-Proximity Joint Index Vs Manipulability Index

In figure 11 it can be seen that manipulators with lower DOF have lower Manipulability Index which makes sense because more DOF gives more movement options to the manipulator.

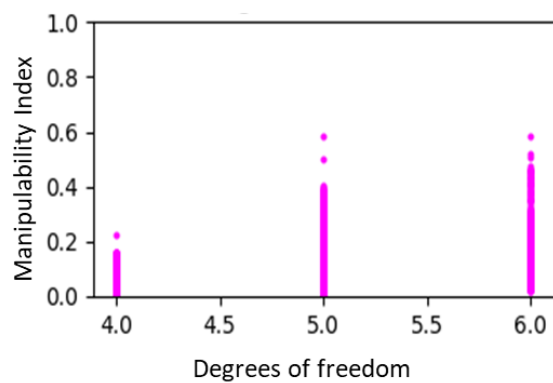


Figure 13 - Manipulability Index Vs DOF

To find If the simulation time is a reliable index to compare between manipulators, 500 configurations were simulated in 2 different computers. In figure 12 it can be seen that sometimes simulation time can be one time high and one time low for the same configuration.

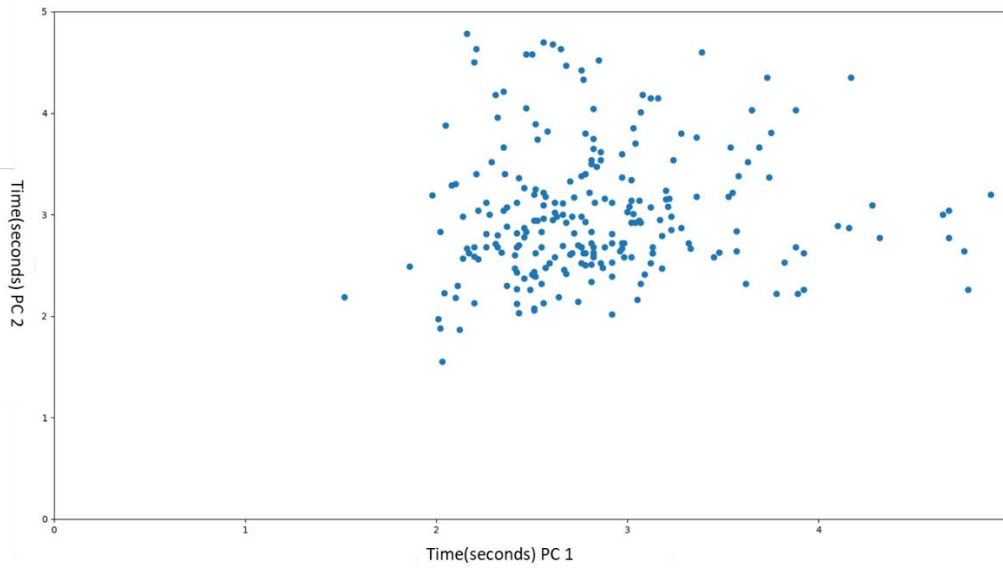


Figure 14 - Time Compassion

4.2. Simulation

4.2.1. Simulator

To find the weight of each link the ratio between the accumulated weight and length of 2 different industrial manipulators. The ratio presented in figure 13. By averaging each component from the two linear equations, the ratio between the accumulated weight and length equation is :

$$Weight[i] = acc_length[i] * 8.79 + 4.29 - acc_weight[i - 1]$$

Where i is ith link number and i = [2,3,4,5,6].

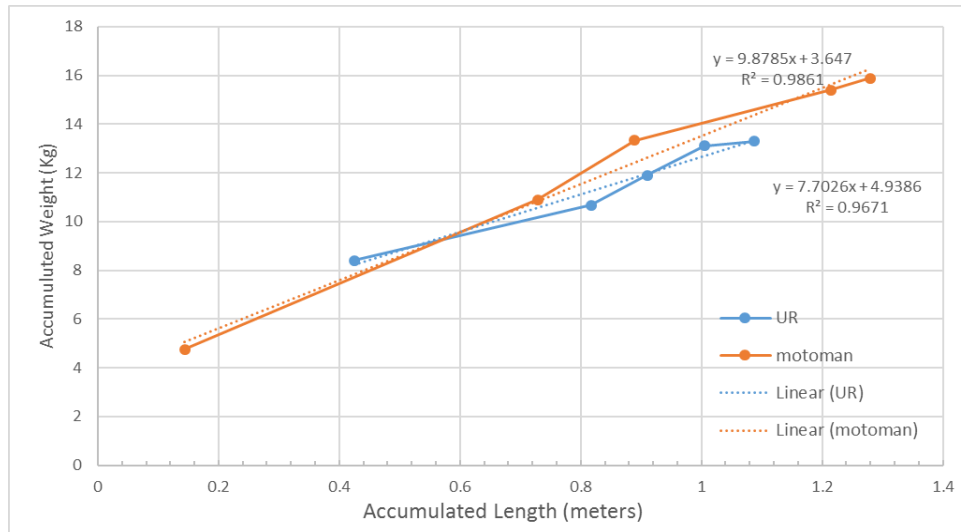


Figure 15 - Accumulated Weight Vs Accumulated Length

4.3. Preliminary Window of Interest

After simulating all the configurations that selected in the configuration selection (section 3.5.1 in methods), simulating all the 4 & 5 DOF configurations, all the concepts with up to 1000 configurations and all the other configurations that were simulated during the build of the algorithms, a Pareto front calculated. In the calculation of the

Pareto front were used about 270,000 configurations, and 216 concepts left to the next step.

The front is the preliminary WOI of evolutionary optimization. In figure 14 it can be seen the preliminary WOI (the yellow surface), the points in the WOI (numbered red dots) and all the other results (blue dots). In table 5 for each point in the WOI its results and the concept, it belongs to. In table 6 for each point assigned the configuration name.

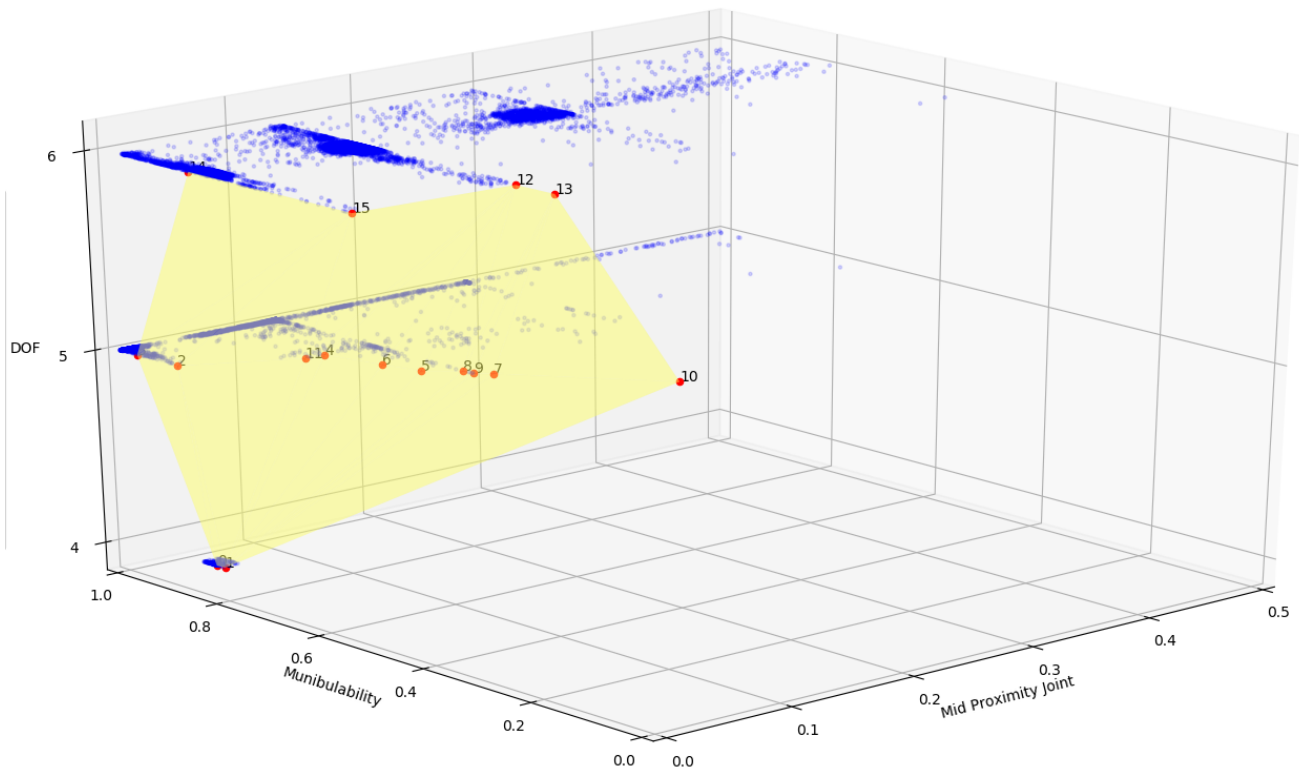


Figure 16 - Preliminary WOI

Table 5 - Concepts in WOI

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# long links	1	1	2	2	0	0	0	0	0	0	0	0	2	1	4	1
longest link	0.7	0.7	0.7	0.7	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.7	0.7	0.7	0.7
DOF	4	4	5	5	5	5	5	5	5	5	5	5	6	6	6	6
Parallel axes about y	0	2	0	0	3	0	0	0	0	0	0	0	2	0	0	2
# pitch joints	2	2	2	1	4	4	2	2	2	2	1	2	5	4	3	5
P/R ratio	0.0	0.0	0.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Acc Length	1.5	2	2	2.6	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	3.1	2	3.6	1.5
mu	0.838	0.824	0.888	0.966	0.813	0.668	0.725	0.599	0.634	0.616	0.416	0.814	0.526	0.457	0.844	0.538
Z	0.006	0.007	0.002	0.001	0.088	0.105	0.098	0.134	0.124	0.125	0.205	0.073	0.123	0.124	0.0	0.001

Table 6 - Configurations in WOI

0	roll_z_0_1pitch_y_0_1pitch_z_0_7roll_z_0_4
1	roll_z_0_1pitch_y_0_4pitch_y_0_4roll_x_0_7
2	roll_z_0_1roll_y_0_7pitch_y_0_7pitch_z_0_1pris_x_0_4
3	roll_z_0_1pitch_y_0_4roll_x_0_4roll_y_0_7roll_y_0_7
4	roll_z_0_1pitch_y_0_4pitch_y_0_4pitch_y_0_4pitch_z_0_1
5	roll_z_0_1pitch_y_0_4pitch_z_0_1pitch_x_0_4pitch_z_0_1
6	roll_z_0_1pitch_y_0_4roll_y_0_4pitch_y_0_1roll_x_0_4
7	roll_z_0_1pitch_y_0_1pitch_z_0_4roll_x_0_4roll_y_0_4
8	roll_z_0_1pitch_y_0_4roll_y_0_1pitch_y_0_4roll_z_0_4
9	roll_z_0_1pitch_y_0_4pitch_x_0_1roll_x_0_4roll_y_0_4
10	roll_z_0_1roll_y_0_4pitch_y_0_1roll_y_0_4roll_y_0_4
11	roll_z_0_1roll_y_0_4roll_y_0_4pitch_y_0_1pitch_x_0_4
12	roll_z_0_1pitch_y_0_7pitch_y_0_4pitch_z_0_4pitch_z_0_4pitch_x_0_7
13	roll_z_0_1roll_y_0_4pitch_y_0_4pitch_x_0_1pitch_y_0_7pitch_x_0_1
14	roll_z_0_1roll_y_0_7pitch_y_0_4pitch_x_0_7pitch_z_0_7roll_y_0_7
15	roll_z_0_1pitch_y_0_1pitch_y_0_1pitch_z_0_7pitch_z_0_4pitch_y_0_1

4.4. Evolutionary optimization with Dynamic-Window of interest

4.4.1. Resource Allocation – Greedy Method

To find T_{high} and T_{low} 6 large concepts were chosen randomly (can be seen in Table 7) and their behavior was studied. In figure 15 it can be seen as an example for one of these runs. The Y-axis is the convergence rate and the X-axis is generation number. it can be seen from the figure that most of the change in the Non-dominated solutions happens in the first 400-500 generations, and if a change made after is very small. From the figure the selected values are: $T_{high} = 0.01$ and $T_{low} = 0.005$.

Table 7 - Checked Concepts

# of long link	Long link	DOF	Par Axes Y	Pitch joints	p/r ratio	Acc length
2	0.7	6	0	2	0	2.6
2	0.7	6	0	4	0	2.6
2	0.7	6	2	4	0	2.6
3	0.7	6	0	4	0	3.1
3	0.7	6	0	3	0	3.1
2	0.7	6	0	3	0	2.6

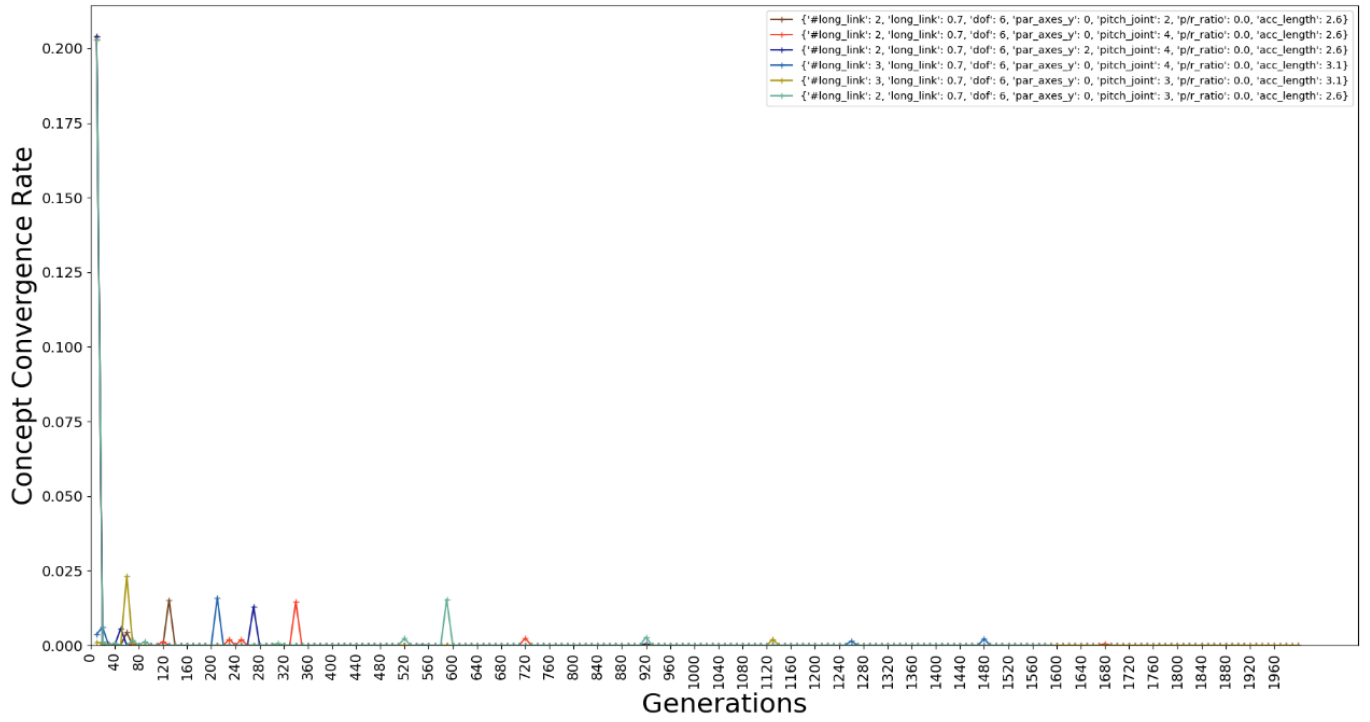


Figure 17 - Greedy method - parameters check

4.4.2. Resource Allocation – Fair Method

5. Conclusions and future work

6. Bibliography

- [1] N. V. Reddy, A. V. V. Reddy, S. Pranavadithya, and J. J. Kumar, "A critical review on agricultural robots," *Int. J. Mech. Eng. Technol.*, vol. 7, no. 4, pp. 183–188, 2016.
- [2] A. Bechar and C. Vigneault, "Agricultural robots for field operations: Concepts and components," *Biosystems Engineering*, vol. 149, pp. 94–111, 2016, doi: 10.1016/j.biosystemseng.2016.06.014.
- [3] C. Wouter Bac and Eldert J. van Henten, J. Hemming, and Y. Edan, "Bac_et_al-2014," *Journal_of_Field_Robotics*, vol. 71, no. 5, pp. 486–494, 2018, doi: 10.1002/rob.
- [4] A. Bechar and C. Vigneault, "Agricultural robots for field operations. Part 2: Operations and systems," *Biosyst. Eng.*, vol. 153, pp. 110–128, 2017, doi: 10.1016/j.biosystemseng.2016.11.004.
- [5] A. Y. C. Nee, "Kinematics of Serial Manipulators," *Handb. Manuf. Eng. Technol.*, no. September 2015, pp. 1–3487, 2015, doi: 10.1007/978-1-4471-4670-4.
- [6] A. Yousuf, W. Lehman, M. A. Mustafa, and M. M. Hayder, "Introducing kinematics with robot operating system (ROS)," *ASEE Annu. Conf. Expo. Conf. Proc.*, vol. 122nd ASEE, no. 122nd ASEE Annual Conference and Exposition: Making Value for Society, 2015, doi: 10.18260/p.24361.
- [7] Y. Kang, D. Kim, and K. Kim, "URDF Generator for Manipulator Robot," *Proc. - 3rd IEEE Int. Conf. Robot. Comput. IRC 2019*, vol. 1, pp. 483–487, 2019, doi: 10.1109/IRC.2019.00101.
- [8] S. Patel and T. Sobh, "Manipulator Performance Measures - A Comprehensive Literature Survey," *J. Intell. Robot. Syst. Theory Appl.*, vol. 77, no. 3–4, pp. 547–570, 2014, doi: 10.1007/s10846-014-0024-y.
- [9] T. Yoshikawa, "MANIPULABILITY OF ROBOTIC MECHANISMS," 1985, pp. 439–446.
- [10] J. K. Salisbury and J. J. Craig, "Articulated Hands: Force Control and Kinematic Issues," *Int. J. Rob. Res.*, vol. 1, no. 1, pp. 4–17, 1982, doi: 10.1177/027836498200100102.
- [11] L. Baron, "A Joint-Limits Avoidance Strategy for Arc-Welding Robots," *Int. Conf. Integr. Des. Manuf. Mech. Eng.*, 2000.
- [12] P. H. Chang, "A Dexterity Measure for the Kinematic Control," 1988, [Online]. Available: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a196223.pdf>.
- [13] C. A. Mattson and A. Messac, "Pareto Frontier Based Concept Uncertainty , with Visualization Selection under Corresponding Author," *Optim. Eng.*, vol. 6, no. 1, pp. 85–115, 2005, doi: 10.1023/B:OPTE.0000048538.35456.45.
- [14] S. Hwang, H. Kim, Y. Choi, K. Shin, and C. Han, "Design optimization method for 7 DOF robot manipulator using performance indices," *Int. J. Precis. Eng. Manuf.*, 2017, doi: 10.1007/s12541-017-0037-0.
- [15] F. Ho and T. Back, "Genetic Algorithms and Evolution Strategies : Similarities and Differences," no. February 1992, 1997.
- [16] P. S. Shiakolas, D. Koladiya, and J. Kebrle, "Optimum robot design based on task specifications using evolutionary techniques and kinematic, dynamic, and structural constraints," *Inverse Probl. Eng.*, 2002, doi: 10.1080/1068276021000004706.

- [17] L. J. Park and C. H. Park, "Application of genetic algorithm to job shop scheduling problems with active schedule constructive crossover," *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 1, pp. 530–535, 1995, doi: 10.1109/icsmc.1995.537816.
- [18] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evol. Comput.*, vol. 13, pp. 501–525, 2005, doi: 10.1162/106365605774666895.
- [19] A. Rosyid, B. El-Khasawneh, and A. Alazzam, "Multi-objective Optimization of planar 3PRR (prismatic-revolute-revolute) parallel mechanism using genetic algorithm," in *Midwest Symposium on Circuits and Systems*, 2017, doi: 10.1109/MWSCAS.2016.7869966.
- [20] K. Deb, *Multi-objective-optimization-using-evolutionary-algorithms*. .
- [21] K. Mahesh, P. Nallagownden, and I. Elamvazuthi, "Advanced Pareto front non-dominated sorting multi-objective particle swarm optimization for optimal placement and sizing of distributed generation," *Energies*, vol. 9, no. 12, 2016, doi: 10.3390/en9120982.
- [22] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]," *IEEE Comput. Intell. Mag.*, 2017, doi: 10.1109/MCI.2017.2742868.
- [23] E. Farhi and A. Moshaiov, "Window-of-interest based multi-objective evolutionary search for satisficing concepts," *2017 IEEE Int. Conf. Syst. Man, Cybern. SMC 2017*, vol. 2017-Janua, pp. 3705–3710, 2017, doi: 10.1109/SMC.2017.8123209.
- [24] B. Samina, "Evolutionary Many Concept Optimization under Multiple Objectives Evolutionary Many Concept Optimization under Multiple Objectives," 2019.
- [25] A. Moshaiov, A. Snir, and B. Samina, "Concept-based evolutionary exploration of design spaces by a resolution-relaxation-pareto approach," *2015 IEEE Congr. Evol. Comput. CEC 2015 - Proc.*, pp. 1845–1852, 2015, doi: 10.1109/CEC.2015.7257111.
- [26] S. S. Rao and P. K. Bhatti, "Optimization in the design and control of robotic manipulators: A survey," 1989. [Online]. Available: <http://mechanicaldesign.asmedigitalcollection.asme.org/pdfaccess.ashx?url=/data/journals/amread/25573/>.
- [27] A. R. Shirazi, M. M. S. Fakhrabadi, and A. Ghanbari, "Optimal design of a 6-DOF parallel manipulator using particle swarm optimization," *Adv. Robot.*, vol. 26, no. 13, pp. 1419–1441, 2012, doi: 10.1080/01691864.2012.690187.