IJRP.ORG
International Journal of Research Publications
ISSN: 2708-3578 (Online)

Reverse Engineering in Robot Design: Classic, GA, AI and DT

PhD Ciprian Dragne[a]

[a] *ciprian_dragne@yahoo.com*
[a] *Institute of Solid Mechanics, Romanian Academy, 010141, Bucharest, România*

Abstract

Reverse engineering (RE) in mechanical studies has many aspects today, and in robotics, it has even more directions and possibilities of study. Such aspects can be included in mechanical studies, hardware, IT (code and control) studies, performance studies, defect and wear studies, identification of mathematical models, systems analysis and their replication studies, and even for educational purposes, etc. A general definition of RE can be considered a return to the initial data of a project, with the information gathered up to the creation of a product, in order to make improvements to it or to understand it more fully, throughout the entire technological development.

This article is a study in which we try to improve the kinematic performance of a serial robot by optimizing the initial values of the parameters (the dimensions of the robotic arms) and to understand more deeply the influence of different kinematic optimization methods for workspaces criteria (displacement workspace, velocity workspace, work tool space) or for the robot's dexterity.

This research presents results obtained by comparing classical, established methods with methods using genetic algorithms (GA), methods using digital twins (DT), methods using artificial intelligence (AI), or methods using criteria from the neutrosophic theory of mathematical logic.

Keywords: Reverse engineering, robotics, dexterity, serial robots, AI, neutrosophic

1. Introduction

Robot reverse engineering (RRE) is a relatively new area of robot application that is most commonly used after a product has been designed. Often, in product design, it is found that continuous improvement is desired to reduce manufacturing costs, to reduce wear and defects, and to improve the performance of these products. The construction of robots has now taken a big leap in development, and they have thus gained more attention from researchers.

The initial requirements in the design of a robot depend on: the constructive configuration (serial, parallel, or mixed configuration robot), the required working space, the operating speed, the loads supported, other operating performances, and, obviously, its costs, etc. Many variables must be taken into account when developing a robot: the size of the robot's workspace, the actuation methods and systems to be used, the robot control applications, the sensors needed to monitor the robot, additional validation or activity tracking systems, etc.

In the case of robotic structures, there is also a strong nonlinearity between all these variables above, if we try to understand some mathematical functions that contain them, especially since, for a practical robot, we are also talking about variables in 3D space. For a serial robot, the control functions (ex: inverse kinematics) in the positioning of the work tool are also functions with nonlinear forms of influence of parameters on the performance of the final robotic arm product There are also strong nonlinear functions between the dimensions of a robot and its operational performance. (Orta, P. et.al. – 2006; Rad, H.- 2012).

The robotic configuration includes not only the global structure of the robot (serial, parallel, mixed) but also the distribution of the kinematic chain depending on the system motors used for actuation (rotary system actuation, linear system actuation), as well as the directions of the local systems of each kinematic joint. This fully defined configuration influences all robot parameters (Dragne, C., & all. - 2021; Dragne, C., & all. - 2022).

We must not forget the performance criteria for increasing the rigidity and strength of robotic arms, as well as for reducing mass, reducing the material used, or other sustainability criteria that have become very important in recent decades due to the problems caused by industry through negative effects on the environment and the habitats of living beings.

3D component printing technologies will allow the creation of new products according to even higher quality standards by maximizing the efficiency of mass production.

For all of the above, we now have another working tool: artificial intelligence (AI), which will allow us to develop new methods for optimizing products and quickly select the most obvious criteria and solutions for improving performance and quality, based on the most practical results previously obtained in design, operation or when repairing returned products.

The main challenges of reverse engineering for robots are:

1. Multiple solutions in robot control: A robot can reach the same final position through several intermediate arm positions or different configurations;
2. Nonlinearity: The relationship between the effector position and the intermediate positions of the actuators at each joint or coupling (complicated mathematical functions);
3. Singularities: There are certain positions where the robot loses a degree of freedom, and the joints become locked or too loose, or the robot requires too high speeds to get out of this state and move further.

After some time in the operation of a robot, some deficiencies are found that deserve attention to be improved, or a new product with higher performance or lower production costs is desired. Determining the best and most efficient robotic configurations must be done after a detailed study that includes as many aspects as possible from those discussed previously.

This study is based on research on the inverse kinematics of a serial robot for two important requirements: positioning a working tool with a certain precision at a point in 3D space and orienting this tool in a required direction. Inverse kinematics (IK) for serial robots is the method that calculates the necessary joint variables (angles or displacements - depending on the chosen constructive configuration for the robot) to place the robot's end effector in the desired position and orientation (robot pose). Unlike forward kinematics, which has only one unique solution, inverse kinematics is very complex, as it often produces multiple solutions, infinite solutions, or no solution at all.

Due to the nonlinearity of the equations for solving the inverse kinematics of a serial robot, finding the best solution requires an iterative solution method.

2. Literature review

In the general design of mechanical structures, there is a very rich literature on mechanical RE (Messler, R. W. – 2014; Stjepandić et all – 2015; Bi, Z., & Wang, X. - 2020). Many of the aspects already discussed above apply in particular to almost all mechanical systems.

But when it comes to robots, the literature is sparser. The first reason is the relative newness of robotic systems, which have only been around for a few decades.

The aspects studied by the new RRE studies are also limited to some specific characteristics of the constructive forms on robotic structures provided by manufacturers, structures which therefore have already fixed configurations (Khalili, A., et all – 2014; Gaz, C., Flacco, F., & all - 2014), or just review the most important aspects (Batni, S., et all - 2010) or specific performances (Dragne, C., & all. - 2021).

There are also studies that reach the criterion of the importance of robots in education (Afari, E., & all – 2017) and as a larger view – concurrent engineering, CAD, IT, internet and interactivity (Dragne, C. – 2025).

The amount of information made available today by the internet is very large and from all corners of the world. AI can also help us in this regard, sorting information discovered from various sources or on the internet, selecting criteria appropriate to us, discovering new methods that we have not heard of or that could not be accessed until a certain point.

Other research in the field of robots has sought to use CAD tools to develop new research methods using all the capabilities made available by a professional CAD program. Thus emerged the idea of a digital twin (DT), a virtual copy of a real system, going as far as a virtual copy in both geometric form and functionality (Dragne, C., & all. – 2022).

All these new research methods will only require more in-depth and longer study time in the early stages. Researchers will have to study more and better. Advanced knowledge of materials science, mechanics, material strength, kinematics, complex robotics, IT, programming, AI, CAD modeling, etc., will have to be required.

Choosing the best programming methods and languages for a robot must also be considered based on the performance required of the robot. The speed of the robot's reaction to detecting certain obstacles, imminent collisions, anomalies, or prohibited areas requires efficient programming in languages and with appropriate hardware (Huei, Y. C. – 2014).

It is important to preview errors and quickly display those performance criteria that are not met, especially if the robot works in very important areas such as medical robots.

I also want to mention here the importance of remodeling, optimizing, and designing new robotic structures in accordance with the newest and most advanced methods, substructures, and materials that have appeared on the supplier market (new motors, meta-materials, new hardware).

Mathematical theorists try to solve the gap between pure mathematics and systems operating in unknown environments. As such, the most recent research has introduced neutrosophical theory.

Neutrosophic theory is a branch of logic that extends traditional fuzzy logic by evaluating specially defined parameters such as truth, falsity, and indeterminacy. In robotics, it is used to allow systems with some autonomy to navigate and make decisions in uncertain, ambiguous, or partially known environments, such as dynamic obstacle avoidance or robot-assisted surgery (Smarandache, F – 2002; Smarandache, F – 2005; Fujita, T., & Smarandache, F. - 2025).

3. Methodology

Are explained here the detailed steps of the methodology. Four different optimization methods will be addressed here:

1. Classic optimization techniques;
2. Genetic algorithm (GA);
3. Artificial intelligence methods (AI);
4. Methods that use digital twin (DT);
5. Hybrid method that combines multiple previous methods.

3.1. Classic optimization techniques

Classical optimization methods are analytical techniques based on differential calculus, used to identify the maximum or minimum values of a function. These methods are suitable for continuous and differentiable functions.

In the case of discrete-valued functions, we can only rely on their continuity. Classical discrete optimization methods find approximate or exact solutions when the decision variables are restricted to specific sets. Unlike continuous optimization, which is based on calculus and gradients, discrete optimization deals with non-differentiable search spaces where function values change by more than certain imposed percentages. These methods can be combined with various other search algorithms for faster search or to systematically narrow the search space (Foulds, L.R. – 1981, Faouri, Adi & all - 2023).

Advantages of classic methods:

- Exact solutions: can find the true best answer using strict math rules;
- Quick implementation in code;
- Clear rules: do not need complex trial-and-error computer runs;
- Recommended for initial testing because each of the other methods used can be more easily developed starting from a classical method;
- Low risk: They use proven formulas that engineers have trusted for many years.

Disadvantages of classic methods:

- Average necessary time for the optimization process;
- Aleatory search of the best solution;
- Search only within a specific range of values;
- A search extension based on objective functions cannot be used if are not implemented intentionally;
- Combinatorial explosion: the solving time increases dramatically with each new design variable;
- Local traps: often get stuck on local peaks, missing the absolute best global design choice;
- Math sensitivity: require smooth, continuous mathematical formulas, which real-world physics rarely offers;
- Rounding errors: rounding continuous solutions to the nearest discrete catalog value can violate critical safety margins.

3.2. Genetic algorithm (GA)

Genetic algorithms (GA) are meta-heuristic algorithms inspired by nature that mimic Charles Darwin's theory of natural selection to evolve a population of candidate solutions over several generations, rather than the classical methods of searching for extremes. While classical methods such as Branch and Bound guarantee finding the mathematically perfect solution, genetic algorithms trade this absolute guarantee for speed and flexibility, making them extremely efficient for massive discrete search spaces or those that are more difficult to understand because of how variables influence the function. But GAs do not work very well with complex functions, are harder to initialize, and the working function must be very clearly defined. In contrast, GA works well with highly nonlinear or black-box objective functions. (Rajeev, S., & all – 1992; Lu, M., & Ye, J. – 2017; Liu, C., et all, - 2021).

Advantages of genetic algorithm methods:
- Global optimization - search multiple design zones at once, skipping local traps to find the absolute best overall solution;
- Can use better criteria for smaller errors;
- No Derivatives Needed: rely only on the final output value, making them perfect for complex, non-linear physics simulations;
- Complex constraints - Multi-Objective Design;
- Parallel processing.

Disadvantages of GA methods:
- High computation cost;
- No Mathematical Guarantee;
- Parameter tuning: finding the right mutation, crossover, and population settings requires extensive trial-and-error guesswork;
- Slow Final Convergence.

3.3. Artificial intelligence methods

AI-based optimization methods for discrete functions use machine learning, neural networks, and probabilistic modeling to navigate vast combinatorial search spaces. While traditional genetic algorithms rely on random, blind mutations, AI-based approaches learn the underlying structure of the problem to predict and generate high-quality discrete choices, but they require prior solutions to train the network. The quality of the results depends on the quality of the reference base, the structure of the neural network, the distribution of layers, and the optimization method implemented (Sun, S., et all, - 2019; Dragne, C., - 2023).

Advantages of AI methods:
- Instant Performance Predictions;
- True Generative Exploration;
- Unbiased Problem Solving: AI operates without human preconceptions, uncovering radical, highly efficient layout alternatives that a traditional engineer might never consider;

Disadvantages of AI methods:
- Lack of safety guarantees – not every reference provides reliable results;
- The “hallucination” problem: AI can produce results that seem OK, but contain other types of errors;
- Data Dependability - AI models are only as good as their training data;
- High Computation and Cost for complex models if are external.

3.4. Methods that use digital twin

Methods based on a digital twin model actually use one of the previous methods, but additionally use the capabilities of a CAD system or program that has implemented complex subroutines for working in 3D space, including behavioral functions in operation.

These additional subroutines can include:
1. Defining components and connections with appropriate constraints (joints, slides, other connection functions);
2. Assessment of the spatial positioning solution of a complex assembly;

3. Positive ordering of components;
4. Collision detection in a real 3D space with all dimensions of the parts;
5. Component parameterization (hence variables for optimization objective functions);
6. High-performance model visualization and animatronics;
7. Scripting or advanced CAX programming, etc. (Dragne, C., - 2024).

Advantages of DT methods:

- Virtual Commissioning & Testing;
- Risk-Free Design Exploration – easier to view and check complex assemblies;
- End-to-end traceability – the model can be included in a broader production management system.
- Quick redesign, updating and correlation of models;
- Extended capabilities for assessment, design optimizations, manufacturing, 3D printing, best material selection, etc.;
- Can be used in assessment of Inverse and Forward Kinematics of robots;
- Can be used in Inverse Kinematics of serial robots or Forward Kinematics of parallel robots, to reach target in only one step.

Disadvantages of DT methods:

- High Infrastructure Costs: deploying a Digital Twin requires expensive specialized software;
- It requires well-trained specialists in multiple fields - mechanics, strength of materials, programming, materials science, production, etc.;
- The "Model Drift" Problem: as a physical machine undergoes real-world wear, rust, or loose tolerance shifts;
- Extreme cybersecurity risks: Connecting different departments to real-time operational data opens severe vulnerabilities for industrial espionage.

3.5. Hybrid method that combines multiple previous methods.

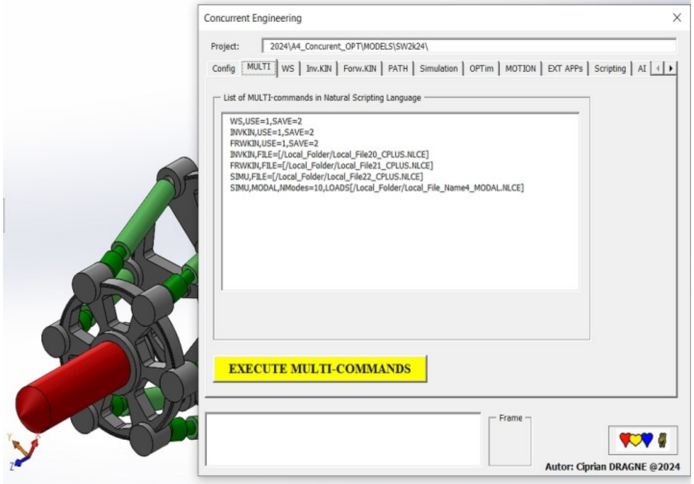
As we saw in subchapter 3.4, we can use a CAD program to solve some problems related to complex modeling in 3D space, including robot behavior functions. But some professional CAD programs (e.g., SOLDIWORKS), also contain scripting subroutines with which we can develop applications that completely control the studied structure, including for its evaluation and optimization, and including with any of the optimization methods presented previously, even with GA or AI.

In Figure 1, can see such a complex application that uses a CAD environment and a DT that is fully controllable, including structural parameterization, evaluation, and optimization, connectivity with other external applications for simulation (e.g., MATLAB) or ARDUINO type for obtaining results from a real experimental model. Figure 1(a) shows the graphical user interface (GUI), and Fig 1(b) shows the list of implemented functionality methods that work with the complex model.

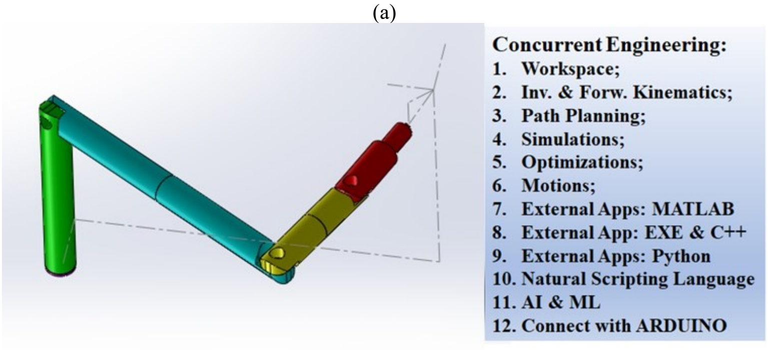
This CAX (Computer-Aided Extended with SW-API) application also has additional:

1. Animatronics functions for presentations;
2. Its own internal programming language for automating the study processes on the CAD models;
3. Additional functions for advanced measurements (projection on 3D space in specific directions);
4. Specifications for assessments using FEA (Finite Elements Analysis) and kinematics inside SOLIDWORKS or with external solvers (CalculiX, Abaqus, etc.);
5. Methods using AI or GA, that can use source codes from internal or external software (C++, VBA, MATLAB or Python);

6. And last but not least, subroutines for the study of robot workspaces, inverse and forward kinematics of robotic structures, and kinematic performances, which we will also use in this article.



(a)



(b)

Fig. 1. Concurrent engineering application integrated into the CAD environment: (a) GUI; (b) List of implemented methods

Geometric and kinematic constraints in CAD programs can be successfully used in solving kinematic problems in the studied structures, collision detection, various distance measurements, more realistic weight evaluation, real geometry study using realistic components, etc., including during movements (Dragne, C., - 2024, July). All these requirements can be achieved sequentially or in parallel (on semi-independent models).

Such applications can be developed for any kind of project, with only some small adaptations for certain specific structures or with additional constraints (e.g., the study of inverse kinematics for a serial robot is done completely differently than the study of inverse kinematics for a parallel robot).

Each of these methods has advantages and disadvantages; some are obvious, others less so.

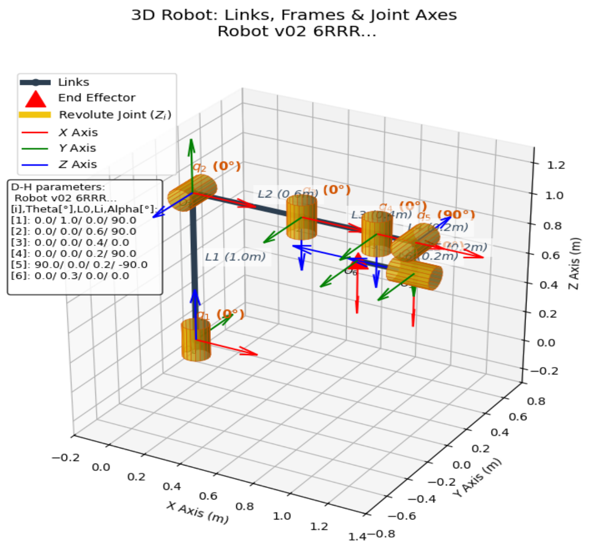


Fig. 2. Robot initial configuration

Figure 2 shows the initial configuration of the 6-degree-of-freedom serial robot used for simulations.

Optimization methods and objective functions have constraints with all 6 degrees of freedom of the tool tip at the target position (location and orientation error).

DT methods work very well in Automated Factories and Robotics by simulating how new robotic arms interact with complex conveyor layouts before setting foot on the factory floor.

There are also major development issues, data integration bottlenecks - merging completely different data formats - such as 3D CAD files, real-time sensor readings and PLM databases for production - creates major problems related to the compatibility of the programs used and the need for massive development of new programs for users.

3.6. Neutrosophic methods in robotics.

Neutrosophy is a branch of logic that extends traditional fuzzy. Unlike standard fuzzy logic (which is based only on membership in a set), neutrosophic logic evaluates any scenario using three components:

- T (Truth)
- I (Indeterminacy), and
- F (Falsity)

Main applications of neutrosophic methods:

1. Robotic motion and joint control: optimize the acceleration and force of robotic joints (especially in humanoid or industrial robots) by evaluating simultaneously multiple objectives, or make decisions in dynamic obstacle avoidance;
2. In Medicine and Surgical Robotics: helps guide delicate maneuvers, such as inserting a biopsy needle or manipulating tissue, or by introducing a mathematical safety margin for handling noisy or ambiguous sensors;
3. Autonomous navigation: allows mobile robots or autonomous cars to plan routes and avoid obstacles when data received from sensors is incomplete or contradictory;

4. Neutrosophic logic can also be extremely effective for solving the inverse kinematics (IK) of a robot, especially when dealing with redundant manipulators, geometric uncertainties, joint clearance, unknown friction, or singularities.

In neutrosophic logic, indeterminacy (I) is the critical component that differentiates this model from traditional fuzzy logic. It mathematically quantifies the unknown, the neutral, the paradox, the missing data, or background noise in a robotic system.

The indeterminacy I value can be a measure of:

1.) Ambiguity:

a. There are differences between a DT model and a model obtained in real time in an operating room or MRI scan if these models overlap and especially in real time;

b. A LiDAR sensor detects an object through dense fog, but cannot distinguish whether it is a solid obstacle or just a cloud of vapor.

2.) Ignorance: the blind spot of a video camera;

3.) Uncertainty in sensor values: the margin of error or electronic "noise" in the data transmitted by a sonar;

4.) or paradoxes: two different error codes or sensor readings that contradict each other in the same millisecond.

How do we manage the value of indeterminacy in a robot to make a correct decision? A robot cannot execute an “indeterminate” command; it must physically act (go around an obstacle, stop, or move forward). To do this, it uses a process called de-neutrosophication by using a scoring function presented in equation (1).

Interpretation of the obtained results:

- The closer the S score is to 1, the “truer” and more reliable the analyzed state is;
- As indeterminacy (I) () or falsity (F) increases, the score decreases, forcing us to introduce safety measures for the robot.

$$\text{Score}, S = \frac{(2 + T - I - F)}{3} \qquad (1)$$

4. Case studies

4.1. Classic optimization techniques

Since the inverse kinematics functions are strongly nonlinear, discrete values are used for optimization, and classical methods based on derivatives and gradients (such as the Hessian matrix or Lagrange multipliers) can no longer be applied, since the functions are no longer continuous.

The method used here is exhaustive search by varying each variable. A simple algorithm is used that tests all options, eliminates those that violate the restrictions, and chooses the best solution.

Only the lengths of the arms L1, L2, and L3 were modified, because the dimensions of the remaining arms (L4, L5, and L6) close to the working tool depend on other design factors. The range of variation of the variables was chosen by multiplying each initial value (L1=1.0, L2=0.6 and L3 =0.4 [m]) with the vector ([0.75, 0.8, 0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15, 1.2, 1.25]).

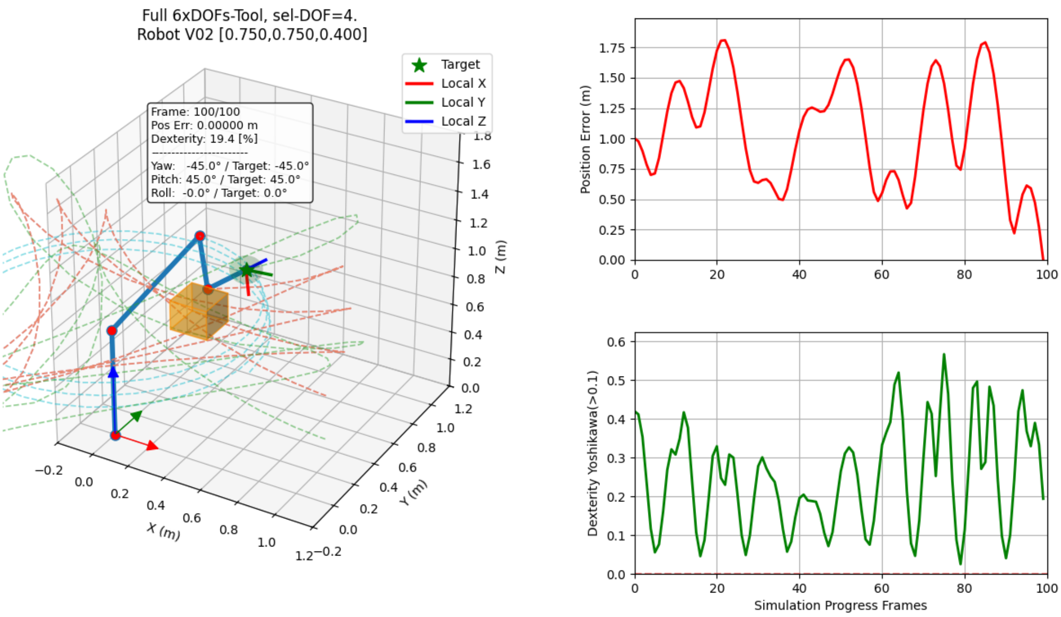


Fig. 3. Optimum robot design using classic method – iterative progress for inverse kinematics

Figure 3 shows the iterative process during simulation only for the optimum configuration with the maximum dexterity of the robot at the target location. The time for each simulation step depends on the search method used, the precision requested, the imposed steps for simulation, the simulation time for plotting the results, and the time for saving data in files.

This classic method can be improved with special techniques that use initial robot location and target location to create an initialization of robot positions for an easier start and evolution of the method simulation. A such method is explained in details in the paper (Dragne, C. – 2019). Before any robotic decision to move to a new position, it's better to evaluate the actual robot position relative to the target, together with any trailing obstacles and possible collisions with other objects and special zones, or even self-collision during the next steps. This method has the name -Directional Kinematics – but is not implemented here yet because of a few factors: even reducing global time for each inverse kinematics search, this method increases time-consuming because it can be applied at each new robot position, and the second factor, we will search for this method in a future paper.

Testing the entire range of parameter variations for the robotic arm will also be a useful task for AI, where the neural network construction and training will be done on all these domains.

4.2. Genetic algorithm (GA)

For the method using genetic algorithms (GA), the link lengths were defined with larger values than in the classical methodology to search for a model with higher dexterity. Table 2 shows parameters for genetic algorithms and two type of results.

Table 1. GA parameters

Parameters of GA	Example 1	Example 2
Number of generations	50	50
Init range [m]	[0.3, 1.5]	[0.3, 1.5]
Solutions per population	8	4
Mutation percent [%]	20	10
Crossover type	Single point	Single point
Best Dexterity [%]	64.3	45.6
Time for optimum search [h]	5	3

Figure 4 shows the result of the optimization using GA, and it can be seen that the required link lengths of the model no.1 are much larger with 64% in dexterity. Now we can understand why a surgical robot like DaVinci has such large arms with many DOFs.

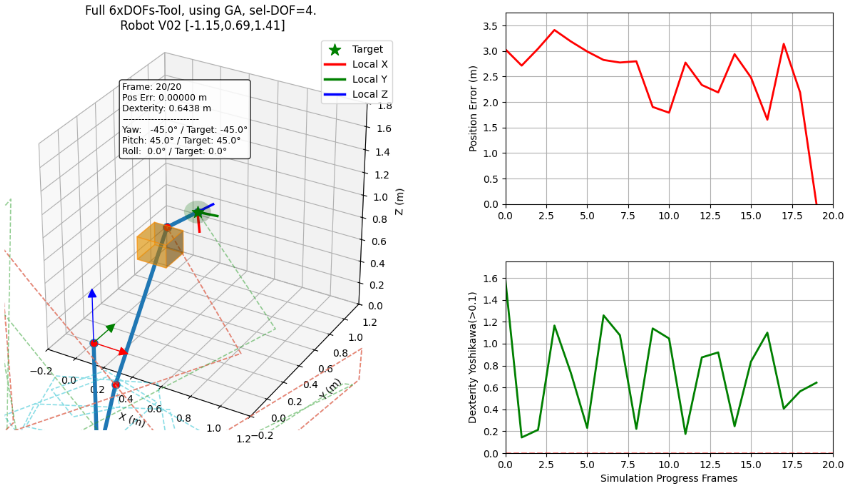


Fig. 4. Optimum robot design using GA method – iterative progress for inverse kinematics for best solution

4.3. Artificial intelligence methods

Instant performance predictions through AI coupled with data reliability (AI models are only as good as their training data) now require a new type of engineer who knows very well how to manage AI platforms or even develop them;

AI-based optimization required an initial database set with previous results for training of AI model. In this test was used an ANN model with 4096 references.

Here was used a Shallow Feedforward Artificial Neural Network (ANN) with 3 Hidden Layers and 32 neurons, how can see in Figure 5. AI model diagnostic profile using MAPE (Mean Absolute Percentage Error evaluation) is presented in Figure 6. MAPE formula is presented in the equation (2).

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\mathrm{Actual(Metric)} - \mathrm{AI(predicted)}}{\mathrm{Actual(Metric)}}\right| \quad (2)$$

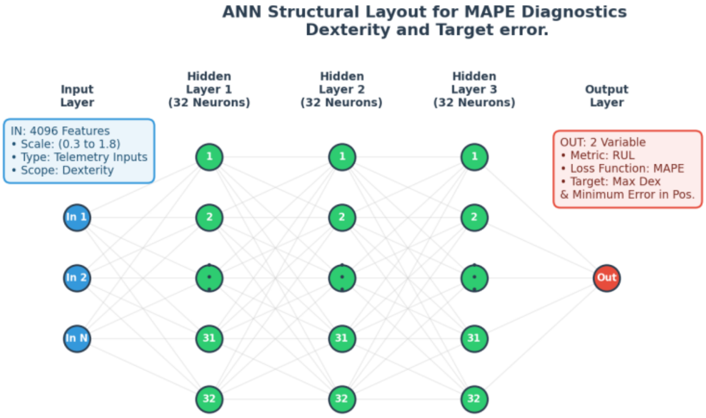


Fig. 5. ANN architecture

The Mean Absolute Percentage Error (MAPE) is one of the most widely used metrics for benchmarking and diagnosing AI models in mechanical engineering.

Advantages of MAPE:

- Its primary advantage is that it evaluates model performance using relative percentages rather than absolute physical units, making it highly intuitive for engineers and business leaders alike;
- Scale independence: Because it measures errors in percentages, you can use MAPE to compare models across entirely different scales;
- High human readability and fast decision - telling a stakeholder that an AI has a "20% MAPE" is universally understood;
- Direct optimization match - deep learning libraries allow you to use MAPE directly as a loss function.

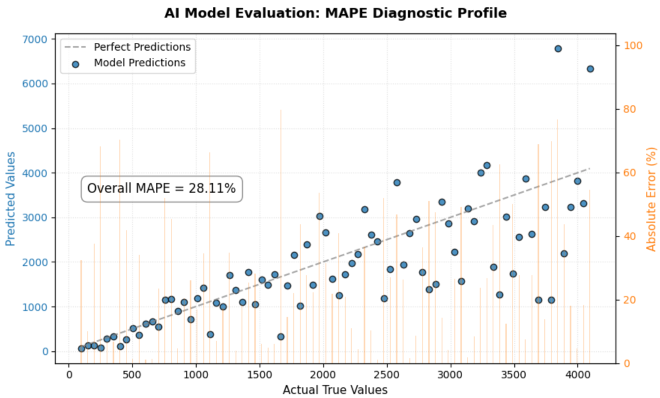


Fig. 6. MAPE diagnostic of ANN

Disadvantages of MAPE:

- The Division-by-Zero trap;
- Asymmetric Penalty Skew;
- Extreme Sensitivity Near Zero (De Myttenaere, A., et all. – 2016; Goodwin, P., & Lawton, R. - 1999).

$$Detection\_Confidence = \max(0,(1-\frac{\sigma^2}{\sigma^2_{max}})\times 100\%) \quad (3)$$

For detection confidence was used equation (3) where σ^2 is prediction variance (Hastie, T., et all – 2009)

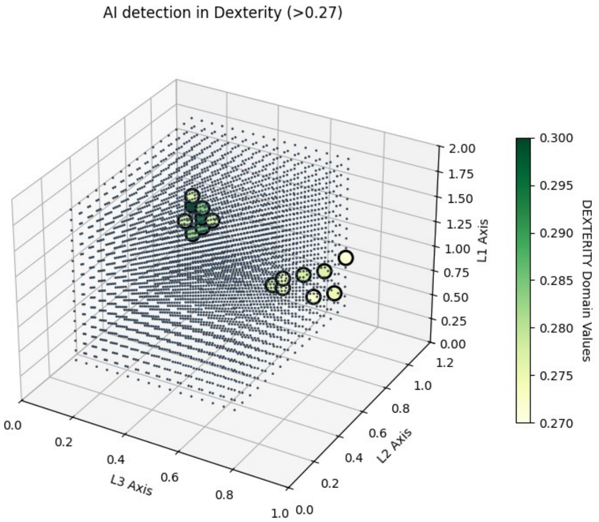


Fig. 7. AI detection

In Figure 7, we can see some AI detections using constraints for dexterity > 0.2 together with the entire database domains. A results example with Inverse kinematic simulation after AI detection is presented in Figure 8. Confidence of the AI detection for the optimum configuration is 92%.

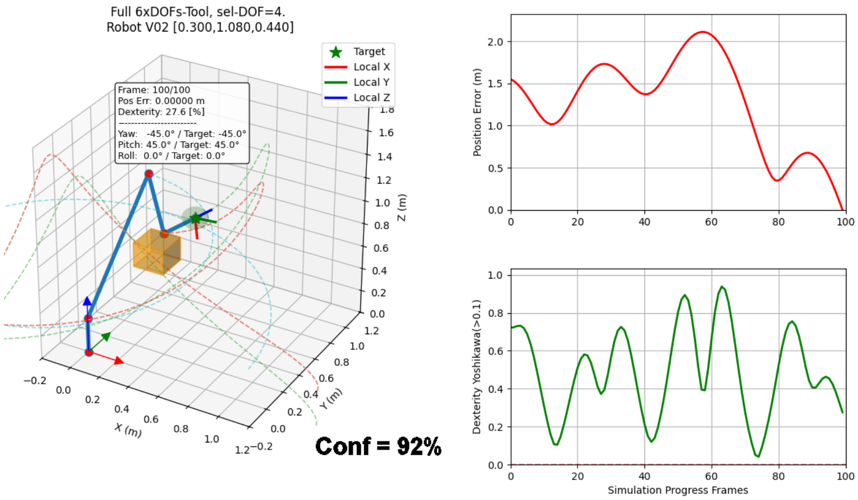


Fig. 8. Simulation of a model obtained after AI detection

More advantages using AI:

- Using references from the dataset for robot movements, additional trajectories can also be evaluated, and movements can be obtained by a combination of multiple sections of different trajectories, etc.;
- Various other criteria can be evaluated very easy: for ex. sustainability criterion in robot movement.

4.4. Methods that use digital twin

Method that uses here DT was developed by the author. As already explained in the Methodology chapter, DT can be used in the inverse kinematics of serial robots or in the forward kinematics of parallel robots to reach the target in a single step. This innovative method uses constraints from CAD programs to precisely position the working tool to the desired position. This movement can be accomplished in a single step.

The control of each robot arm is done using a CAX-type application developed in the SOLIDWORKS API. The CAD model is parameterizable and also controls the geometric dimensions of the robot.

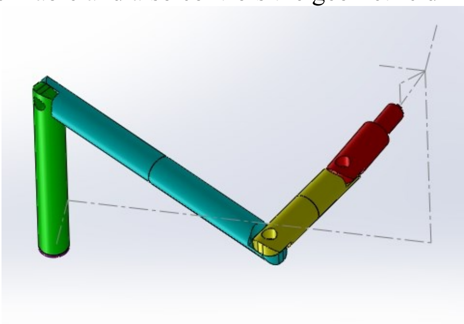


Fig. 9. A typical results using optimization with DT

Figure 9 shows an optimum CAD model using optimization with DT. Obtained length of links are $L_{1,2,3}$ = [0.75, 0.8, 0.4] and dexterity is 20.4 %.

More advantages using DT:
- A complete 3D CAD model that can contain many other necessary constructive elements: motors with their real geometry, additional elements needed for a robot (video cameras), sensors in their realistic geometric shape, etc.;
- Collision detection assessment in realistic mode;
- Model dynamics and durability assessment using professional CAD applications.

More disadvantages using DT:
- Hard to develop such CAX application for most engineers;
- Complicated to manage DOFs of robots and CAD constraints in the same time.

4.5. Hybrid method that combines multiple previous methods.

A typical framework for using multiple methods in robot design is presented in Figure 10. The optimization sequence can be different from how it is presented in this figure.

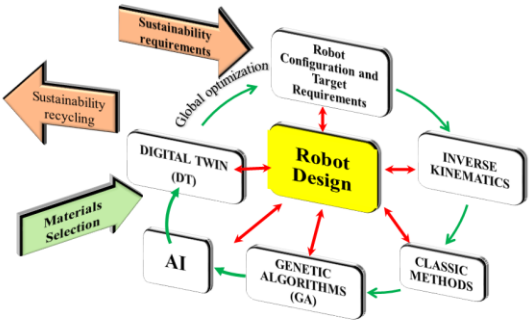


Fig. 10. Framework for multiple methods in robot design

When combined, these four methods create a continuous loop of design, testing, and real-time enhancement.

A typical workflow example might contain the following recommendations:
1. CLASSIC METHODS – assessment in Inverse Kinematics, defines physics boundaries, stress limits, model dynamics, and CAD baselines;
2. GENETIC ALGORITHMS (GA) - swaps link lengths and part shapes to maximize workspace size;
3. ARTIFICIAL INTELLIGENCE (AI) - replaces slow math solvers to predict real-time component stress, collision detection and obstacle avoidance;
4. DIGITAL TWIN (DT) – validates designs virtually, advanced assessment in durability and sustainability requirements, and even updates the physics model.

This complex methodology can operate as a complete cycle or each method is partially involved in different stages, depending on the user's objectives and level of knowledge.

4.6. Neutrosophic methods in robotics

This subchapter was added here as an introduction to neutrosophic methods that can be used in robotics design. As was already explained, the neutrosophic logic introduces a third independent metric: Indeterminacy (I). In robotics, ”I” represents unknown, neutral, or unmeasurable real variables - such as navigating in unfamiliar environments, erratic slippage, sensor signal noise, little-known friction, or gaps in a robot's joint.

Neutrosophic methods can also be used in robot path optimization, where instead of treating unknown readings as solid obstacles, engineers use neutrosophic multicriteria decision-making (MCDM) techniques, such as neutrosophic PROMETHEE or WASPAS. The robot calculates the path with the absolute lowest cost, simultaneously minimizing both the risk of falsehood and the score of indeterminate blindness.

Advantages of Neutrosophic methods:

- Complete Independence of Parameters - its unique ability to isolate and manage Indeterminacy (I) independently from Truth (T) and Falsity (F);
- Smoother Actuator and Dynamic Control - when used in robotic velocity or joint control (such as the Robot Neutrosophic Control method), it eliminates the jerky, erratic movements;
- Robust Multi-Objective Compromises;
- Handles Incomplete, Contradictory Data (Smarandache, F. – 2005).

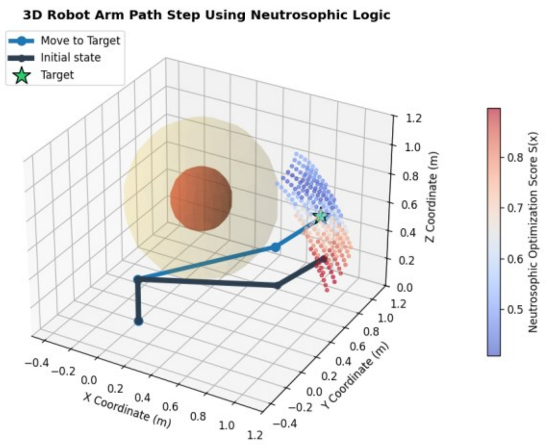


Fig. 11. Neutrosophic virtual test

Figure 11 shows a neutrosophic virtual test that require an obstacle avoidance. Obstacle avoidance methods allow a robot to safely navigate its environment by detecting and avoiding static or even moving obstacles. These methods are broadly classified into local (reactive) methods, which handle immediate, real-time adjustments based on sensor data, and global (map-based) methods, which plan an optimal trajectory from start to finish using a known map. The research details for this method will be explained in a future paper.

Disadvantages of Neutrosophic methods:

- High computational complexity because must track and calculate independent mathematical distributions for T, I, and F;
- The Paradox of Independence: because neutrosophical logic treats Truth, False, and Indeterminacy as completely independent variables, this could lead to counter-intuitive or paradoxical results;
- Arbitrary de-neutrosophization when the trivalent neutrosophic matrix must to be converted back to a single clear scalar value.

4.7. Special case: extended optimization of robot configuration.

The possibility of further optimizing the robot through larger changes to the configuration (DOF order and Denavit-Hartenberg parameters) was studied here. Starting from the optimal model obtained with the help of artificial intelligence, a classical optimization method was implemented for the Denavit-Hartenberg parameters related to alpha angles.

Figure 12 shows four dexterity optimum cases with variations also in the Alpha angles. The time required for the simulation was more than 24 hours.

The best dexterity model obtained so far is:

- Links length [m]= [L1, L2, L3, L4, L5, L6 = 0.3, 1.0, 0.45, 0.1, 0.1, 0.1]
- ALPHA angles [deg] = [a1, a2, a3, a4, a5, a6 = 30, 60, 90, 45, -45, 0]
- Best Dexterity = 35.64%

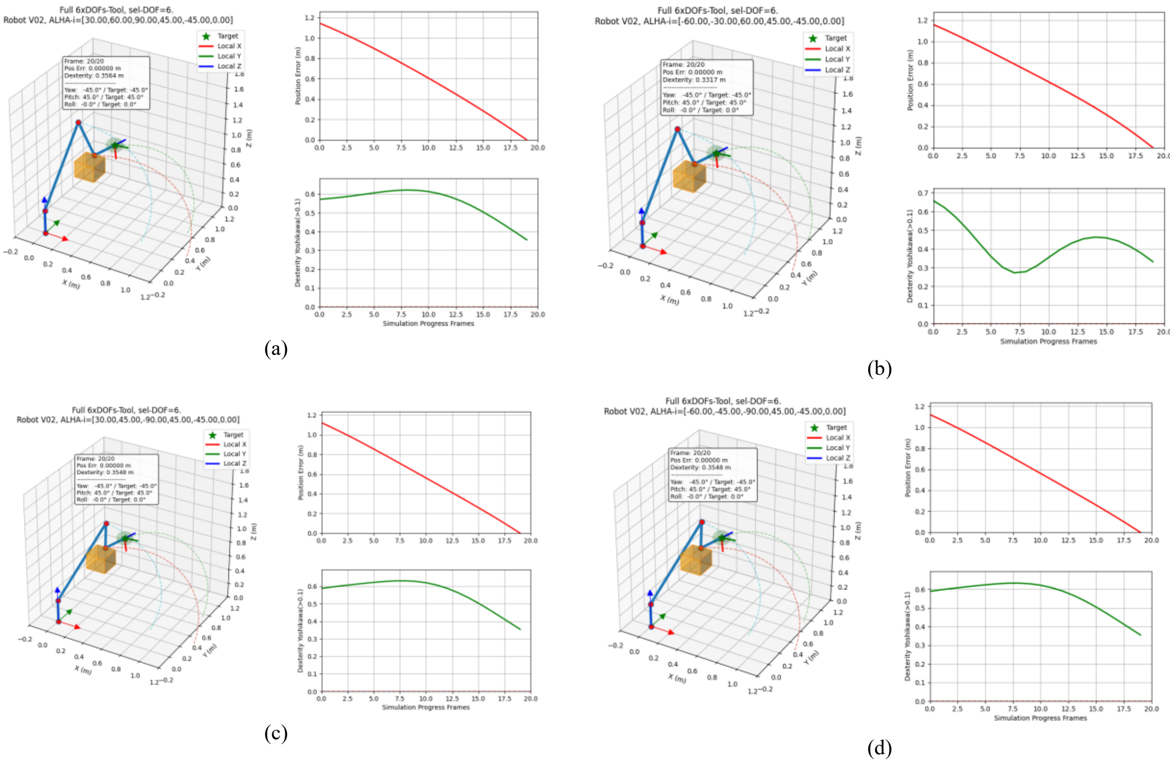


Fig. 12. Optimization in Denavit-Hartenberg parameters (ALPHA angles)

In Figure 13 are presented all last models with Dexterity higher than 32%.

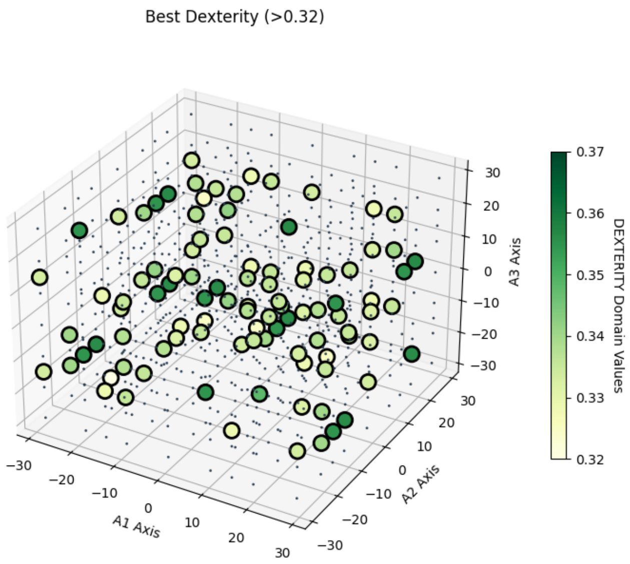


Fig. 13. Models with Dexterity > 32%

5. Conclusions

Modern engineering methods no longer need to treat classical, genetic, artificial intelligence, and digital twin methods as rivals. Instead, they need to combine them into a single continuous loop, taking advantage of each method in areas sensitive to the others.

Such a combination can use artificial intelligence for raw speed, genetic algorithms for global depth exploration, classical methods for strict safety validation, and digital twins to better validate the model in terms of strength or mobility criteria in 3D space, or to connect everything to the real world.

General conclusion of this paper:

1. The main conclusion is that, today, it is more efficient to use multiple optimization methods for the same design, not necessarily for the same areas or criteria of interest, but as a chain of methods that work together to obtain a better product from all required points of view;
2. Each method has advantages and disadvantages. The choice of method sequences must take into account the initial requirements and the costs of design, production, and recycling;
3. As some methods are innovative, so are the applications. Professional CAD and CAE programs will incorporate more and more evaluation methods over time. At this point, interested users and researchers must develop the necessary applications themselves;
4. I will not give a clear verdict here on the most effective method. In the end, the best method is the one we can implement best;
5. New emerging technologies (AI and Neutrosophic) will require increasingly in-depth studies by researchers;
6. The most effective method in terms of results obtained for dexterity proved to be the GA method;
7. The fastest method was with AI, but only because there was an initial database.

Recommendations in using methods:
1. It is recommended to always start with classical methods. This way, we will learn to develop step by step;
2. Some methods need a huge amount of time for calculations for the most common requirements in precision. Start with smaller values of parameters for errors;
3. Read the recommended articles first with a pencil in hand (to understand equations) before you start implementing them yourself;
4. Choose the programming languages you know best; don't try to develop with unknown programming languages because it will take you a long time to develop applications.

This paper wad developed using Python, SOLIDWORKS, and other author codes and resources.

Future research:
1. Study for more complex interactive applications for robots;
2. Improved GUI for visualization and results;
3. Interconnectivity with more external applications;
4. More research using combined multiple methods.

Acknowledgements

This paper was developed by PhD Ciprian Dragne, București, România, Research Engineer level 3 (AFER). Actual work place Institute of Solid Mechanics – Romanian Academy.

Declaration of conflicting interests

The author declares that has no conflicts of interest. This research was supported by the author's own funds.

References

Afari, E., & Khine, M. S. (2017). Robotics as an educational tool: Impact of Lego Mindstorms. International Journal of Information and Education Technology, 7(6), 437-442.
Bi, Z., & Wang, X. (2020). Computer aided design and manufacturing. John Wiley & Sons.
Khalili, A., Natale, L., & Tacchella, A. (2014). Reverse engineering of middleware for verification of robot control architectures. In International Conference on Simulation, Modeling, and Programming for Autonomous Robots (pp. 315-326). Cham: Springer International Publishing.
Batni, S., Jain, M. L., & Tiwari, A. (2010). Reverse engineering: a brief review. International Journal on Emerging Technologies, 1 (2), 73-76.
Dempere, L. A. (2009). Reverse engineering as an educational tool for sustainability. In 2009 IEEE International Symposium on Sustainable Systems and Technology (pp. 1-3). IEEE.
De Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2016). Mean absolute percentage error for regression models. Neurocomputing, 192, 38-48.
Dragne, C. (2019). Directional Kinematics—First Step in Robotic Movement. Analele Universităţii "Eftimie Murgu,
Dragne, C., & all. (2021). Hexapod workspaces–positions, speeds, forces. In The 9th International Conference on Computational Mechanics and Virtual Engineering, Comec-2021 (pp. 21-23).
Dragne, C., & all. (2022). Mechanical engineering of robotic systems by SOLIDWORKS. International Journal of Modern Manufacturing Technologies (IJMMT), 14(2).

Dragne, C., (2023), AI in Mechanical Design: Generate Specific Components, 4th International Conference on Computational Intelligence (ICCI-2023), Surat, India, indexed in Proceeding Springer ICCI 2023, Chapter 15 in book Algorithms for Intelligent Systems, Proceeding Springer ICCI-2023, ISBN 978-981-97-3525-9, DOI:10.1007/978-981-97-3526-6

Dragne, C., (2024). Advanced Concurrent Engineering using CAD software - Classic, Avant-garde and AI. International Journal of Advanced Multidisciplinary Research and Studies, July 17, DOI: https://doi.org/10.62225/2583049X.2024.4.4.3038.

Dragne, C., (2024, July). Using Geometric Constraints in Robot Kinematics. In 2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC) (pp. 1290-1296). IEEE.

Dragne, C. (2025). Super Digital-Twins for Academic Research. International Journal of Scientific Advances, ISSN, 2708-7972.

Faouri, Adi & Kasap, Pelin. (2023). Classical optimization techniques, https://www.researchgate.net/publication/378635305_CLASSICAL_OPTIMIZATION_TECHNIQUES

Foulds, L.R. (1981). Classical Optimization. In: Optimization Techniques. Undergraduate Texts in Mathematics. Springer, New York, NY. https://doi.org/10.1007/978-1-4613-9458-7_7.

Gaz, C., Flacco, F., & De Luca, A. (2014). Identifying the dynamic model used by the KUKA LWR: A reverse engineering approach. In 2014 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1386-1392). IEEE.

Goodwin, P., & Lawton, R. (1999). On the asymmetry of the symmetric MAPE. International journal of forecasting, 15(4), 405-408.

GRABCAD author codes, https://grabcad.com/phd.ciprian.dragne-1, last accessed 2026/05/28.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning (2nd ed.). Springer.

Huei, Y. C. (2014). Benefits and introduction to Python programming for fresh more students using inexpensive robots. In 2014 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE) (pp. 12-17). IEEE.

Kellogg, R.S; Jenison, R., (1997), “Utilizing Reverse Engineering to Explore the Design Process”, Section 2438, 1997 ASEE Annual Conference & Exposition, June 15-18, Milwaukie, Wisconsin.

Liu, C., Wu, W. Z., & Xie, W. (2021). Study of the generalized discrete grey polynomial model based on the quantum genetic algorithm. The Journal of Supercomputing, 77(10), 11288-11309.

Lu, M., & Ye, J. (2017). Guided genetic algorithm for dome optimization against instability with discrete variables. Journal of Constructional Steel Research, 139, 149-156.

Messler, R. W. (2014). Reverse engineering: mechanisms, structures, systems, and materials. New York, Chicago, San Francisco (USA): McGraw-Hill Education.

Orta, P., et.al., (2006), “Use of Reverse Engineering as a Teaching Tools in Mechanical Engineering Education”, Section 2546, 2006 ASEE Annual Conference & Exposition, June 18-21, Chicago, Illinois.

Otto, K.N.; Wood, K.L., (1996), “A reverse Engineering and Redesign Methodology for Product Evolution”, Proceedings of the 1996 ASME Design Engineering Technical Conferences and Design Theory and Methodology Conference, August 18-22.

Parlak, S., Tokel, N., & ÇAKIROĞLU, Ü. (2024). Reverse engineering in robotics classrooms: Boosting creative thinking and problem solving. International Journal of Computer Science Education in Schools, 7(1).

PYTHON Homepage, https://www.python.org/, last accessed 2026/03/22.

Rad, H. (2012, June). Reverse engineering as a learning tool in design process. In 2012 ASEE Annual Conference & Exposition (pp. 25-1130).

Rajeev, S., & Krishnamoorthy, C. S. (1992). Discrete optimization of structures using genetic algorithms. Journal of structural engineering, 118(5), 1233-1250.

Smarandache, F. (2002). Book. Neutrosophy, a new Branch of Philosophy. Infinite Study.

Smarandache, F. (2005). Book. A unifying field in logics: neutrosophic logic. Neutrosophy, neutrosophic set, neutrosophic probability: neutrsophic logic. Neutrosophy, neutrosophic set, neutrosophic probability. Infinite Study.

Fujita, T., & Smarandache, F. (2025). Book. Reconsideration of neutrosophic social science and neutrosophic phenomenology with non-classical logic. Infinite Study.

SOLIDWORKS Homepage, https://www.solidworks.com, last accessed 2026/05/12.

Stjepandić, J., Wognum, N., & Verhagen, W. J. (2015). Concurrent engineering in the 21st century. Foundations, developments and challenges. Springer.

Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2019). A survey of optimization methods from a machine learning perspective. IEEE transactions on cybernetics, 50(8), 3668-3681.