KINEMATIC-BASED STRUCTURAL OPTIMIZATION OF ROBOTS

[1] **Hazim Nasir GHAFIL**[*] [2] **Károly JÁRMAI**

[1,2] Faculty of Mechanical Engineering and Informatics
Institute of Energy Engineering and Chemical Machinery, University of Miskolc
Miskolc, H-3515 Miskolc, Egyetemváros, Hungary
e-mail: [1]vegyhnr@uni-miskolc.hu, [2]altjar@uni-miskolc.hu
[1] University of Kufa, Najaf, Iraq

Abstract: In this paper a detailed description of a method is presented to estimate the minimum structural dimensions of the robot arms. A comparative study is conducted between the harmony search and artificial bee colony algorithms in this scientific application. The comparison process was done through the kinematic equations of the serial robot manipulator to find the optimum lengths of links of the robot. A novel design for a seven-degrees-of-freedom robot arm was presented to conduct the comparative study on the presented optimization algorithms. This novel robot mimics the functionality of the SANDVIK robot arm for tunnelling works, but the presented type synthesis was designed to overcome the restrictions on the original SANDVIK arm.

Keywords: Optimization, Structural optimization, Harmony search algorithm, Artificial bee colony, Robotics, Dimensional synthesis

1. Introduction

In the kinematic combination of a multi-linkage robot arm, topology optimization is performed to obtain the best type synthesis [1] for a single task or multitask robot manipulator. Next, the minimum required dimensions should be calculated during the dimensional synthesis process [2]. Both the type and dimensional synthesis can be done in two separated operations or one single operation. A genetic algorithm has been employed [3] to find the optimum topology and dimensions of a planar linkage depending on graph theory definition. A graph theory-based approach has been used [4]

---

[*] Corresponding Author

to estimate the type synthesis while the best dimension parameters are calculated by the precision position method. On the dimensional level, shape optimization of the link of the robot using level set techniques have been dealt with in [5]. On the type level, the critical point in most works is the graph theory [6], where possible topologies are enumerated to get the best type. Various optimization techniques can be employed to solve different problems like traveling salesman [7], stiffened plates [8] or heat exchangers [9].

In this work, harmony search and artificial bee colony optimization algorithms were compared to investigate their efficiency to solve the structural optimization problem of robot manipulators. A specific topology of robot arm was used to find the dimensional synthesis depending on the kinematic equations. A robot arm with novel topology and 7 degrees of freedom (7 DOF) was designed and employed to estimate the best dimensionality for multitasking purposes. A comparative study also has been done between the artificial bee colony and harmony search algorithm. The test function for this comparison is the kinematic equations of the manipulator mentioned above, and this test function of the ten-dimensional length is hard enough to be a challenge to the optimization algorithms to reach to the global minimum. The new type of robot arm which is proposed in this paper is designed to overcome restrictions on the arm SANDVIK DD421 [10].

2. Harmony search

The behavior of the musicians in a jazz ensemble during improvisation of a new harmony first inspired the Harmony Search (HS) algorithm [11]. HS is a robust heuristic optimization algorithm [12]. The following procedure can be used to solve a problem:

1. Define problem parameters like the limits of the search space, number of variables, and the objective function;
2. Define parameters called Harmony Memory Consideration Rate (HMCR), HMCR=0.95, and Pitch Adjusting Rate (PAR), PAR=0.1;
3. Initialize a random matrix (Harmony Memory (HM)), where each row represents a different solution while each column represents a component (variable) in the solution;
4. Evaluate each row (solution) by calculating the cost for it;
5. Initialize a random number and if it is less than HMCR then compose a new solution from random rows and columns, if not then create a random solution within the search space;
6. Initialize a random number and if it is less than PAR then shift the components of the new solution towards its neighbors slightly;
7. Evaluate the new solution and if it is better than the worst solution in HM then use it to replace the worst solution in HM.
8. Repeat step 5 until reaching a stop condition.

The main parameters of the harmony search algorithm are given in *Table I*.

3. Artificial bee colony

This is a metaheuristic algorithm belonging to swarm intelligence and was inspired by the foraging behavior of honey bees [13]. The artificial bee colony ABC is efficient and returns perfect solutions for many optimization problems. The procedure to solve a problem is given in detail in the following:

1. The random population of candidate solutions has to be initialized randomly;
2. Employed bee section: For each solution in the population calculate the neighborhood structure using

$$sol_i = sol_i + \phi_i(sol_i - sol_k),\tag{1}$$

where sol denotes a particular solution, i is the current iteration number, ϕ is a random number $\phi_i \in [-1,1]$, and k is another random number $k \in [0,1]$.

3. Onlooker bee section: The probability of each solution has to be calculated according to its fitness value

$$f_i = \begin{cases} \dfrac{1}{1+O_i}, & O_i \geq 0, \\ 1 + abs(O_i), & O_i < 0, \end{cases}\tag{2}$$

$$P_i = \frac{f_i}{\sum f_i},\tag{3}$$

where O_i is the objective value for any solution; f denotes fitness value; and P is the probability of the current solution in the iteration. Neighborhood structure should be applied to the solutions as mentioned in section 2.

4. Scout bee section: If any solution does not get better after a moderated time or number of iterations, it should be replaced by a random solution within the search space. The following pseudo-code can demonstrated the ABC algorithm.

Algorithm Pseudo-code of Artificial bee colony

```
Initialization:
Initialize the random population of solutions;
Calculate the initial cost function value, f(Sol);
Set best solution, Solbest ← Sol;
Set Abandonment rate for each solution in the population, A ← 0;
Set maximum number of iterations, NumOfIte;
Set the population size;
// where population size = OnlookerBee = EmployeedBee;
Set Control limit CL ← OnlookerBee*2;
iteration ← 0;
```

Improvement:
do while (iteration < *NumOfIte*)
  for i=1: *EmployeedBee*
      Evaluate random number $\phi_i \in [-1, 1]$, *k*;

Apply the neighborhood structure on each solution in the population
Sort the solutions in ascending order based on the penalty cost;
Sol* ← select the solution that has the best value;
      if (*Sol** < *Solbest*)
      Solbest = Sol*;
      else A_i**++**
      end if
Determine the probability for each solution, based on the following:

$$f_i = \begin{cases} \dfrac{1}{1+O_i}, & O_i \geq 0, \\[2mm] 1 + abs(O_i), & O_i < 0, \end{cases}$$

$$P_i = \frac{f_i}{\sum f_i},$$

end for
  for i=1: *OnlookerBee*
    Sol* ← select the solution, which has the higher probability;
      Evaluate random number $\phi_i \in [-1, 1]$, *k*;

      Apply the neighborhood structure on each solution in the population
      Sort the solutions in ascending order based on the penalty cost;
      Sol* ← select the solution that has the best value;
      if (*Sol** < *Solbest*)
      Solbest = Sol*;
      else A_i++
      end if

      for i=1: population size
      if (A_i >= CL)
      Set random solution instead of the current solution;
      A_i ← 0
      end if
iteration ++
end do

The main parameters of the Artificial bee colony algorithm are given in *Table II*. The parameters of the HS and ABC algorithm have been chosen unfairly, the function evaluation FE for HS is less than FE for ABC. However, harmony search gives the best results, as it will be explained later.

Table I

Harmony search parameters

Parameter	Value
Iterations	5,000
Harmony memory size	25
Harmony memory consideration rate	0.9
Pitch Adjustment Rate	0.1

Table II

Artificial bee colony parameters

Parameter	Value
Iterations	1,000
Number of population	300
Onlooker bees	150
Employed bees	150
Limit	6,750,000

4. Structural optimization of 7 DOF type

In this paper, a massive hydraulic tunneling manipulator SANDVIK DD421 (*Fig. 1*) was taken into consideration to inspire a new design of robot manipulator that overcomes the restrictions on this model of SANDVIK. The proposed topology of the robot should enable the arm to work in a confined environment which is a real challenge for designers. The most prominent examples of a confined space are found in tunneling and mining works, especially when the requirements are digging a narrow or low overhead tunnel. Also, the problem gets more complicated in case of manipulation of the rubble as well as drilling the hole or pit.

Fig. 2 and *Table III* illustrate the type synthesis and Denavit-Hartenberg parameter respectively of the new robot, which is designed to meet the requirements mentioned above; all of the variables are dimensionless. It is assumed that a_2, a_3, a_4, a_5 are equal, and for this problem, ten variables represent a single task point. *Fig. 3* presents a given trajectory that should be followed by the robot end-effector with its optimum dimensions. For each point in the trajectory, there should be optimal structure, and finally, the enumeration should be applied to all the proposed structures to choose the best option for reaching all of the task points in the working space.

The homogeneous transformation matrices will be estimated from the Denavit matrix and Denavit parameters in *Table III* as follows:

$$t_1 = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

Fig. 1. SANDVIK DD421 robot
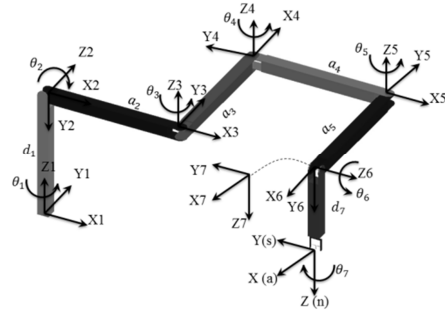manipulator [10] (with permission)

Fig. 2. Frame assignment of the proposed
7 DOF robot manipulator

Table III

Denavit parameters for the proposed 7DOF robot

$Link_i$	α_i	a_i	d_i	θ_i
1	-90	0	d_1	θ_1
2	90	a_2	0	θ_2
3	0	a_3	0	$90+\theta_3$
4	0	a_4	0	$90-\theta_4$
5	-90	a_5	0	$90-\theta_5$
6	-90	0	0	θ_6
7	0	0	d_7	θ_7

$$t_2 = \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) & a\cos(\theta_2) \\ \sin(\theta_2) & 0 & -\cos(\theta_2) & a\sin(\theta_2) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

$$t_3 = \begin{bmatrix} \cos(\theta_3+90) & -\sin(\theta_3+90) & 0 & a\cos(\theta_3+90) \\ \sin(\theta_3+90) & \cos(\theta_3+90) & 0 & a\sin(\theta_3+90) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

$$t_4 = \begin{bmatrix} \cos(\theta_4-90) & -\sin(\theta_4-90) & 0 & a\cos(\theta_4-90) \\ \sin(\theta_4-90) & \cos(\theta_4-90) & 0 & a\sin(\theta_4-90) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{7}$$

$$t_5 = \begin{bmatrix} \cos(\theta_5 - 90) & 0 & -\sin(\theta_5 - 90) & a\cos(\theta_5 - 90) \\ \sin(\theta_5 - 90) & 0 & \cos(\theta_5 - 90) & a\sin(\theta_5 - 90) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

$$t_6 = \begin{bmatrix} \cos(\theta_6) & 0 & -\sin(\theta_6) & 0 \\ \sin(\theta_6) & 0 & \cos(\theta_6) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{9}$$

$$t_7 = \begin{bmatrix} \cos(\theta_7) & -\sin(\theta_7) & 0 & 0 \\ \sin(\theta_7) & \cos(\theta_7) & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

The total transformation matrix *tot*, which relates the end-effect or to the base frame will be

$$tot = t_1 \cdot t_2 \cdot t_3 \cdot t_4 \cdot t_5 \cdot t_6 \cdot t_7. \tag{11}$$

The overall matrix *tot* is 4×4 dimensions and the index of x, y, and z are represented as $x = tot(1,4)$, $y = tot(2,4)$, and $z = tot(3,4)$; they are not written here due to lack of space. Thus, the objective function for this optimization problem will be

$$obj = \sqrt{(x - x_t)^2 + (y - y_t)^2 + (z - z_t)^2} + penalty, \tag{12}$$

where the subscript t denotes task point coordinates and the penalty is a high number like 100 or more; it is added to the objective function if and only if any point belonging to the robot arm touch the boundaries of the workspace, which are denoted as constraints in *Fig. 3*. A random 100 task points are continuously distributed in the workspace, and during the enumeration process, the best structure - one that can achieve all points - should be chosen. For any point among the random 100, there should be a robot configuration that can be represented by ten kinematic variables. The ten variables are seven for joint angles (θ_1, θ_2, θ_3, θ_4, θ_5, θ_6, θ_7) plus three for the length of the link d_1, a, and d_7. These ten variables are substituted in equations (4)-(11) to get x, y, and z coordinates to be replaced in the objective function (12). The lower and upper limits for each variable are shown as follows:

$-160 \leq \theta_1 \leq 160$, $-150 \leq \theta_2 \leq 30$, $-120 \leq \theta_3 \leq 120$, $-45 \leq \theta_4 \leq 130$,

$-30 \leq \theta_5 \leq 180$, $-90 \leq \theta_6 \leq 90$, $-100 \leq \theta_7 \leq 100$,

$500\,\text{mm} \leq d_1 \leq 1000\,\text{mm}$, $300\,\text{mm} \leq a \leq 700\,\text{mm}$, $200\,\text{mm} \leq d_7 \leq 400\,\text{mm}$.
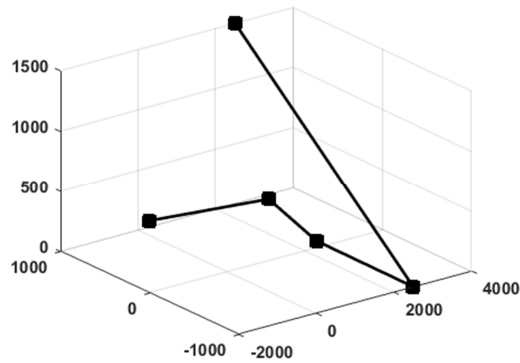
Fig 3. The trajectory of the given problem

Fig. 4 expresses five possible configurations, each corresponding to a single point on the trajectory. Both the HS and ABC algorithms return their solution for the best dimensional synthesis of the robot that satisfies all points on the given trajectory.
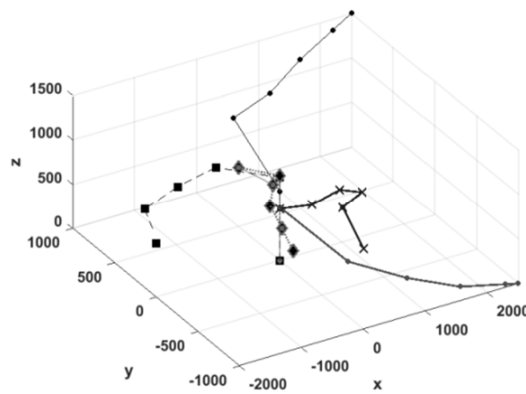

Fig. 4. Different configurations for the 7 DOF manipulator, each configuration with different line style to make it easy to notice; nodes refer to the joints of the robot while the boundaries of the space are the constraints which have to handle the cost function

The expected result is that the optimization algorithm creates a configuration that reaches the task point. In other words, the optimized joint angles and lengths of links should converge the objective function to zero. As it is shown in *Table IV*, harmony search performed the job well and converged to zero, while Artificial bee colony could not go to zero and stuck at an objective value equal to 4.7094. Though the artificial bee colony is sufficient for most optimization problems, it starts showing a reduction in its efficiency when there are many decision variables, as it is confirmed by this problem (ten variables).

The novel 7 DOF robot manipulator with its optimized dimensions is simulated in a 3D environment as it is shown in *Fig. 5*; this was created using the 3Ds Max software

package. The constrained workspace matches the space in *Fig. 3*, and the simulated model worked well in its virtual environment.

Table IV

Structural components of the 7 DOF robot arm

Link mm	Harmony search	Artificial bee colony
d_1	855.8	1,102.2
a	508.3	777.1
d_7	307.9	191.2
Objective value	0	4.7094
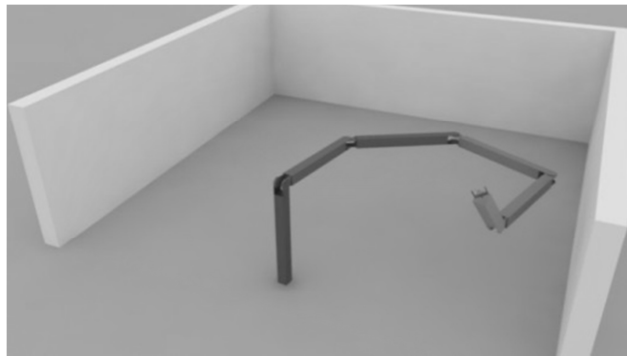
Fig. 5. 3D simulation of the 7 DOF robot manipulator

5. Conclusion

By following the rules of the structural optimization and dimensional synthesis, the cost of the components can be reduced to the minimum value, which is important for economic and environmental reasons. Kinematic equations of robot manipulators are one of the quantities that can be optimized to reach the optimum structure of the robot arm. While there are many optimization algorithms, a comparative study has been done between harmony search and artificial bee colony because they are classified as effective optimization methods. The comparison process was made by optimizing the structural components of a novel 7 DOF robot manipulator. According to many tests, harmony search regularly drives the objective value to zero, which is the optimum value, but artificial bee colony algorithm failed to converge to the global of the function even with more function evaluation than HS. Another aspect of this study is that it presents a novel design for a multitask robot arm for tunneling and mining works.

Acknowledgements

development of the University of Miskolc aiming at intelligent specialization' project implemented in the framework of the Széchenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund.

Open Access statement

References

[1]   Kong X., Gosselin C. Type synthesis of single-loop kinematic chains, in: *Type synthesis of parallel mechanisms*, Springer, Vol. 33. 2007, pp. 43–53.

[2]   Kivelä T., Mattila J., Puura J. A generic method to optimize a redundant serial robotic manipulator's structure, *Automation in Construction*, Vol. 81, 2017, pp. 172–179.

[3]   Liu Y., McPhee J. Automated kinematic synthesis of planar mechanisms with revolute joints, *Mechanics Based Design of Structures and Machines*, *An Int. J,*Vol. 35, No. 4, 2007, pp. 405–445.

[4]   Pucheta M. A., Cardona A. Topological and dimensional synthesis of planar linkages for multiple kinematic tasks, *Multibody System Dynamics*, Vol. 29, No. 2, 2013, pp. 189–211.

[5]   Tromme E., Tortorelli D., Brüls O., Duysinx P. Structural optimization of multibody system components described using level set techniques, *Structural and Multidisciplinary Optimization*, Vol. 52, No. 5, 2015, pp. 959–971.

[6]   Kawamoto A., Bendsøe M. P., Sigmund O. Planar articulated mechanism design by graph theoretical enumeration, *Structural and Multidisciplinary Optimization*, Vol. 27, No. 4, 2004, pp. 295–299.

[7]   Kota L., Jarmai K. Efficient algorithms for optimization of objects and systems, *Pollack Periodica*, Vol. 9, No. 1, 2014, pp. 121–132.

[8]   Virág Z. Optimum design of stiffened plates, *Pollack Periodica*, Vol. 1, No. 1, 2006, pp. 77–92.

[9]   Varga T., Szepesi G., Siménfalvi Z. Horizontal scraped surface heat exchanger - Experimental measurements and numerical analysis, *Pollack Periodica*, Vol. 12, No. 1, 2017, pp. 107–122.

[10]   *Sandvik_manual*, DD421 Development drill rig, https://www.rocktechnology.sandvik/ globalassets/products/underground-drill-rigs-and-bolters/pdf/dd421-specification-sheet-english.pdf, (last visited 14 October 2018).

[11]   Geem Z. W., Kim J. H., Loganathan G. V. A new heuristic optimization algorithm: Harmony search, *Simulation*, Vol. 76, No. 2, 2001, pp. 60–68.

[12]   Ghafil H. N. Inverse acceleration solution for robot manipulators using harmony search algorithm, *Int. J. Computer Applications,* Vol. 144, No. 6, 2016, pp. 1–7.

[13]   Karaboga D. An idea based on honey bee swarm for numerical optimization, *Technical Report-TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.