Dexterity Optimization of a Three Degrees of Freedom DELTA Parallel Manipulator

Vitor Gaspar Silva, Mahmoud Tavakoli, and Lino Marques

Department of Electrical and Computer Engineering,
Institute of Systems and Robotics, University of Coimbra, Portugal
v.gaspar.silva@gmail.com, {mahmoud,lino}@isr.uc.pt

Abstract. This paper demonstrates dexterity optimization of a Delta-like three degrees of freedom (3 DOF) spatial parallel manipulator. The parallel manipulator consists of three identical chains and is able to move on all three translational axes. In order to optimize the manipulator in term of dexterity, a Genetic Algorithm (GA) global search method was applied. This algorithm aims to propose the best design parameters such as the length of the links which results in a better dexterity. Results of the optimization are presented.

1 Introduction

Parallel Manipulators are closed-loop mechanical systems, used on the industry and many other activities of high technical demand like aerospace, for their good performances in terms of accuracy, rigidity, manipulation speed and ability to manipulate large loads [1,2,3]. One of the first parallel manipulator was developed by Stewart [4]. The Stewart platform is a 6 DOF. manipulator with 6 arms. Despite the many advantages of parallel mechanisms, they also has several drawbacks [2,5], demonstrate some disadvantages that are shared by many parallel manipulators: complex direct kinematics; position and orientation of the moving platform are coupled; small workspace; expensive joints, namely the spherical. Other platforms were proposed to tackle some of these disadvantages. The DELTA platform designed by Clavel [6] solves the first two items. Nonetheless, it is a 3-4 DOF manipulator. Delta manipulators are popular manipulators that are mainly being used for pick and place applications. Their main advantage is their moving speed within their workspace. Figure 1 shows a 3 DOF version of the Delta robot, which was constructed in our lab, and the schematic of its kinematic chain. An advantage of delta robots is that the links a and b can be easily extended or shortened, while keeping the main platform and its actuators unchanged. In this way one can change the workspace of the robot for a desired task. However changing these parameters has also an effect on the dexterity of the robot. In contrast with serial articulated arms, where singularities are located on the border of the workspace, parallel manipulators suffer from singularities inside their workspace, and thus some areas in the workspace of the manipulator should be avoided [7]. Dexterity of a manipulator may be viewed

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_51, © Springer International Publishing Switzerland 2014

as a degree of farness or distance from a singularity. On the other hand the location of such singularities in the workspace changes with changing length of the links of the manipulator. It is desired to choose the best design parameters in order to increase the dexterity of the manipulator. Considering the complex nature of the parallel manipulators kinematics, global search methods are usually applied for optimization of the dexterity of the parallel manipulators [8]. Stamper et al. [9], proposed a method for optimization of a three degree of freedom translational parallel manipulator both total workspace and global conditioning index. The total workspace of the manipulator is determined by a Monte Carlo technique. The optimization of the manipulator design for the global condition index results in a manipulator where the lower leg comprises 44% of the total leg length and the upper arm comprises 56% of the total leg length, while legs are installed on a circle with an angular separation of $120'$. They showed that optimization of the total workspace volume results in significantly different design parameters compared to the optimization performed for a well conditioned workspace. Moreover, these results showed that a manipulator of this type that is designed to maximize total workspace volume will result in an ill conditioned workspace. Tavakoli et. al. considered optimization of the design parameter for a multi objective object function which comprises of both dexterity and workspace volume [10] using a genetic algorithm method. This work addresses the problem of optimizing the dexterity of this spatial three degrees of freedom parallel manipulator with a Genetic Algorithm method.

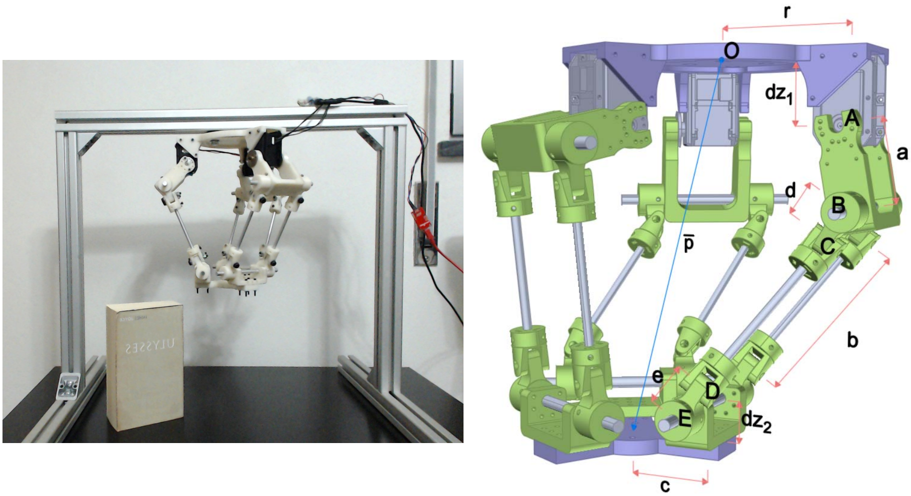

Fig. 1. The spatial three degrees of freedom parallel manipulator and schematics of its kinematics chain

2 Optimization

Dexterity of a manipulator may be viewed as a degree of farness or distance from a singularity. To have a dexterity measurement of a parallel manipulator the Global Conditioning Index (GCI) represented by η in equation 1 is used. This was first proposed in [11] and expands the condition number [12] of the manipulator's to the entire workspace W.

$$\eta = \int_W \frac{1}{\gamma} dW \tag{1}$$

Where γ is the *condition number*, which defines the amplification factor that the error has on the end-effector velocity, using the inverse kinematics Jacobian (J). This number is dependent on the norm, which in our case is the Euclidean norm in the 3D space. The smallest possible value of the condition number is 1, meaning that the error is not amplified. The inverse of this value is used, $s = 1/\gamma$, to transform the problem in a maximization one in the domain of [0 1].

$$\gamma = \|J^{-1}\|\|J\| \tag{2}$$

Some limitations on this condition number approach are pointed in [1]. The most significant is that if the manipulator has rotational DOF on the Cartesian space the number does not translate in a clear physical meaning. They are instead transformed on "equivalent" translations. As the analyzed manipulator has only translational DOF no additional methods are to be taken into account. Now that the condition number is defined for one point, there is the need to expand this to the whole workspace in order to evaluate the manipulator. The integration of a manipulators workspace is complex, hence a numeric method was used. The Monte Carlo method was selected, since it has been successfully applied in similar tasks [9,13]. The Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. In this work, a known workspace is defined and points are randomly generated according to a uniform distribution. Those points are tested on the kinematic function of the manipulator to check if they rely on the manipulator's workspace. Finally the relation between the number of points tested and the number of points that are valid on the manipulator workspace, provides the relation between the known workspace and the manipulator workspace. Using the Monte Carlo method, a number of points is created inside a volume that is confined by half of a sphere on the positive Z axis. This sphere has a maximum theoretical radius limited by the sum of the manipulator parameters $[a+b+d+e]$ (Figure 1). The created points are then checked if they are on the workspace, and, if so, their condition number is calculated. The volume of a sphere is given by $V = \frac{4\pi r^3}{3}$. Considering only half of the sphere and the proportion of the points on the Monte Carlo method where the n_{valid} are the points inside the workspace, where $r_s = a + b + d + e$ is the sphere radius limited by the manipulator links lengths. The sum of all valid points will have a relation to the workspace volume.

Thus the global conditioning index calculated with the Monte Carlo method for the parallel manipulator in study is defined by:

$$GCI = \frac{2\pi {r_s}^3 \sum_i \frac{1}{\gamma}}{3n_{total}} \quad \text{for } i \text{ valid points.} \tag{3}$$

In this formula the term $\sum_i \frac{1}{\gamma}$ sums the condition number of the "i" valid points. When it is divided by "n" total points, it results in the ratio of valid points to total points. If this ratio is multiplied by the initial considered volume of the half sphere, it results in an estimation of the actual workspace volume of the manipulator.

It is worth mentioning that the GCI, relates to the volume of the workspace, presenting a normalised value for all lengths of the arms and workspaces. That is, if for instance, the sum of the lengths of the links are considered as 1, the actual volume of the workspace can be avoided in the formulation. Dexterity depends on the length of a parallel manipulator's links. Since these manipulators contain a large number of links and their configuration imposes dependencies between the links, the study of the optimal configuration to achieve a better dexterity in the workspace is a complex problem, which is usually studied by global search algorithms such as Genetic Algorithm.

2.1 Genetic Algorithm

In order to optimize the dexterity of the manipulator within its workspace Genetic Algorithms (GAs) based search was applied. GAs based on natural evolution are a search heuristic commonly used to find a solution for complex problems, and have been used with success across several problem domains [14]. In our work it was built a floating point GA to optimize the GCI, where the length of the links compose the genome of the individuals presented in the population. In floating point GA method, the traditional binary representation of the solutions is substituted by real numbers. Thus mutation and cross over functions are different with the binary and most common GA optimization methods.

To guarantee convergence and avoid local maxima, the operators that guide the search have to be tailored to the encoding of the solution and the origin of the problem. In the binary GA, the mutator function is defined based on swapping one or more binary digits of each chromosome. In floating point GA methodology, the mutation function is performed by a gaussian mutator. On the gaussian mutator a gaussian function with a specified standard deviation is applied to the allele, that way is possible to produce a new value within a certain range of the old value. This operator has an important role in the search on the nearest neighbourhood. Due to the nature of the problem and the fact that the mutation is so tailored to the problem the crossover operator will promote the population diversity, so it may be used without regarding the structure of the genome, making two new individuals from the recombination of two parents' genome.

It was ran some initial simulations in order to tune the parameters of the GA in order to guarantee convergence of the solutions. As a result it was initiated a population of 150 individuals through 250 generations. In each generation, individuals are selected in a tournament of 5 to generate the next generation, these individuals have 0.9 probability to be crossbreed and 0.2 probability to be mutated. The next generation is composed by the crossbreed and mutated individuals and 2% of the best individuals of the previous generation (Elitism operator in Figure 2). It was also introduced a novel operator, called armageddon. When the best solution does not suffer alteration on 10 consecutive generations the operator armageddon replaces 92% of the population with new individuals, creating new diversity on the search space. Figure 2 shows the structure of the GA built for the optimization of the GCI of the manipulator.

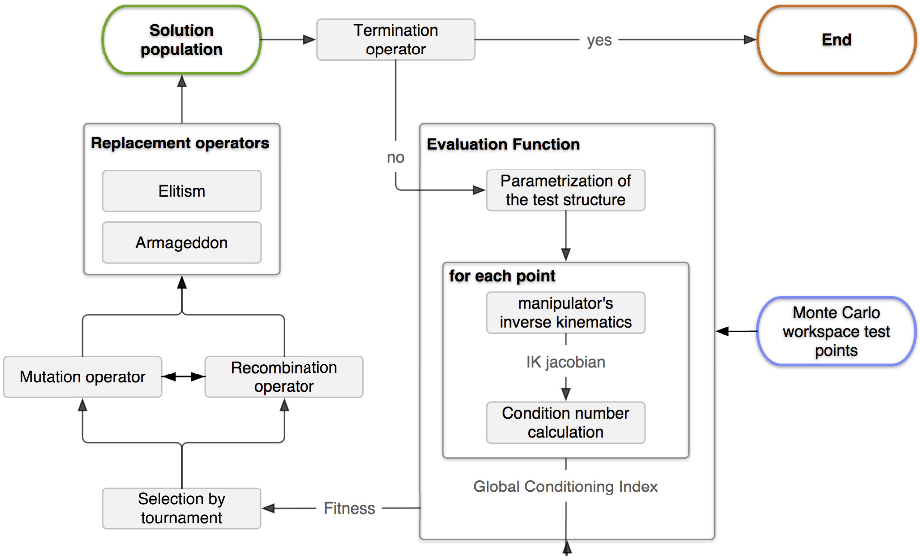

Fig. 2. Genetic Algorithm Implementation

3 Results

After the tuning the operators that compose the GA and guaranteeing its convergence, 30 runs of the algorithm each with 250 generations were made in order to have a statistic validation of the results, allowing to verify the validity of the algorithm solution to the optimization. In the Monte Carlo method 100000 test points were considered.

Table 1 shows the fitness of the GA to the problem. in our approach this means the GCI of the possible structures created by the GA. Table 1 shows the

Table 1. Test 1 solutions summary of the 30 runs

	Mean	**Median**	**Max**	**Min**	**Std**
a	0.31682	0.31620	0.39609	0.23653	0.05354
b	0.66159	0.65047	0.74893	0.58653	0.05527
d	0.00533	0.00233	0.03226	0.00000	0.00994
e	0.00748	0.00326	0.02066	0.00000	0.00854
c	0.39494	0.43838	0.74988	0.00000	0.26823
r	0.43295	0.51821	0.76442	0.00000	0.27619
fitness	0.09407	0.09482	0.09966	0.08400	0.00457

structure parameters of the last generation in each run representing the statistic parameters of the best solutions. Table 2 shows the best solution obtained for the Test 1. Based on the fitness low standard deviation (0.00457), in table 1, and the high standard deviation (0.26823) on c and (0.27619) r, on table 1, it can conclude that parameters c and r do not have a direct relation with the manipulator's GCI. The values for the parameters d and e are close to 0, with a low standard deviation.

Table 2. Tests' best solutions

	a	**b**	**d**	**e**	**c**	**r**	**c-r**
Test 1	0.35440	0.62724	0.01023	0.00813	0.74988	0.76442	
Test 2	0.34689	0.65170	0.00065	0.00075			0.00178
Test 3	0.34506	0.65494					

Therefore as a first result it was considered $e = d = 0$. Also as it can be seen that c and r values are not directly connected to the improvement of the manipulator's GCI, and thus in the next test it was considered "the difference between c and r" as a solution parameter. Based on the results (Table 2), it can be seen that for a better GCI the $c - r$ should be 0. The focus will then be on the parameters a and b, considering $c = r = 1$, and $d = e = 0$.

Therefore it was ran the optimization for the third time with the results of the previous two steps. The results obtained confirm the ones obtained in the previous tests with $a = 0.34506$ and $b = 0.65494$. With the lowest standard deviation (0.00748) on these parameters on the tests conducted as expected. Meaning that for our symmetric manipulator the link length $a = 35\%$ of the total length of the each arm, and $b = 65\%$ is the optimal.

As shown on previous tests, in order to improve the GCI of the manipulator the structure parameters d, e, difference $c-r$ must be 0. Allowing us to set those values, to better explore the impact the structure parameters a and b have on the manipulator's GCI, as done in the third test. But as this is done the complexity of the problem decreases. So it is now possible to apply a "brute force" method

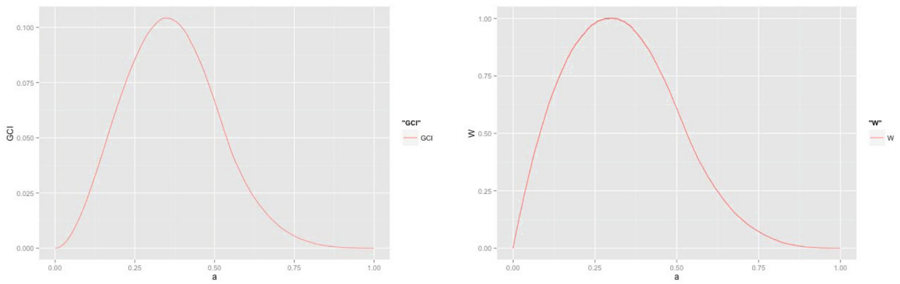

Fig. 3. Variation of the GCI (left) and the workspace volume (right) against the length of the link "a"

exploring all the combinations of a and b. As imposed before, the length of the arm composed by $a + b + d + e$ is 1. This means that for this test $b = 1 - a$.

Figure 3, shows the evolution of the GCI by its value S as a function of a. The GCI maximum value is reached when the manipulator configuration as the structure parameters: $a = 0.35102$; $b = 0.64898$. The evolution of the workspace volume is shown in Figure 3, validating our previous results with the GA.

4 Conclusions

The evolutionary algorithm was developed and validated, and with it the optimization of the dexterity of the manipulator's workspace was performed. This optimization showed a structure configuration of the manipulator where: the length of the link $a = 35\%$ of the total arm's length; $b = 65\%$ of the total length; the lengths d and e equal to 0; and the difference between the length of the base and manipulator $r - c = 0$. It could also derive the effect of length of each parameter on the workspace volume and dexterity of the manipulator (Figure 3).

References

1. Merlet, J.P.: Parallel Robots. Solid Mechanics and Its Applications, vol. 128. Springer, Heidelberg (2006)
2. Pierrot, F., Fournier, A., Dauchex, P.: Towards a fully-parallel 6 dof robot for high-speed applications. Robotics and Automation (April 1991)
3. Gosselin, C.: A new architecture of planar three-degree-of-freedom parallel manipulator. Robotics and Automation, 3738–3743 (April 1996)
4. Stewart, D.: A platform with six degrees of freedom. Proceedings of the Institution of Mechanical Engineers 180(1), 371–386 (1965)
5. Hunt, K.H.: Structural Kinematics of In-Parallel-Actuated Robot-Arms. Journal of Mechanisms Transmissions and Automation in Design 105 (1983)
6. Clavel, R.: DELTA, a fast robot with parallel geometry. In: Burckhardt, C.W. (ed.) Proc of the 18th International Symposium on Industrial Robots, pp. 91–100. Springer, New York (1988)

7. Agrawal, S.: Workspace boundaries of in-parallel manipulator systems. In: Fifth International Conference on Advanced Robotics, Robots in Unstructured Environments, ICAR 1991, vol. 2, pp. 1147–1152 (1991)
8. Gallant, M., Boudreau, R.: The synthesis of planar parallel manipulators with prismatic joints for an optimal, singularity-free workspace. Journal of Robotic Systems 19(1), 13–24 (2002)
9. Stamper, R., Tsai, L.W., Walsh, G.: Optimization of a three dof translational platform for well-conditioned workspace. In: Proceedings of the1997 IEEE International Conference on Robotics and Automation, vol. 4, pp. 3250–3255 (April 1997)
10. Tavakoli, M., Zakerzadeh, M., Vossoughi, G., Bagheri, S., Salarieh, H.: A novel serial/parallel pole climbing/manipulating robot: Design, kinematic analysis and workspace optimization with genetic algorithm. In: 21st International Symposium on Automation and Robotics in Construction, Korea (2004)
11. Gosselin, C., Angeles, J.: A global performance index for the kinematic optimization of robotic manipulators. Journal of Mechanical Design 113, 220 (1991)
12. Salisbury, J.K., Craig, J.J.: Articulated hands force control and kinematic issues. The International Journal of Robotics Research 1(1), 4–17 (1982)
13. Alciatore, D., Ng, C.: Determining manipulator workspace boundaries using the monte carlo method and least squares segmentation. In: 23rd ASME Mechanisms Conference, pp. 141–146. American Society of Mechanical Engineers, Minneapolis (1994)
14. Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G.: Genetic programming IV: Routine human-competitive machine intelligence. Springer (2005)