DESIGN OF THE TWO-SERIAL HEXAPOD OF DISCRETE MANIPULATOR

Roche Alimin
Department of Mechanical Engineering
Petra Christian University
Surabaya, East Java 60236, Indonesia
ralimin@petra.ac.id

Felix Pasila, *IEEE Member*
Department of Electrical Engineering
Petra Christian University
Surabaya, East Java 60236, Indonesia
felix@petra.ac.id

***Abstract* – In this paper, a novel two-serial Hexapod of 3D dicrete manipulator (double hexapod) is developed. The designing of a parallel manipulator with actuators, which controlled discretely, is a must because the mechanism uses artificial intelligence method when dealing with the inverse static analysis (ISA) problem. In this approach, mathematical model of the manipulator is not compulsory.**

The research method used simulation software modeling and testing with the case of two serial-hexapod of discrete manipulator with 12 actuators.

A neuro-fuzzy network type Takagi-Sugeno (NFTS) with Levenberg-Marquardt Algorithm (LMA) is proposed as ISA solution, where three displacements and twelve discrete actuators as inputs and outputs of network respectively. The performance of NFTS shows that the error, such as Root Mean Squared Error, is 3.39% in average.

Index Terms – two-serial hexapod, parallel discrete manipulator, Inverse Static Analysis, NFTS.

I. INTRODUCTION

Parallel manipulator is a manipulator that consists of a number of actuators which are arranged in parallel. In general, parallel manipulator mechanism consists of a combination of several joint, where the actuators that move the manipulator serves as a Prismatic joint. Parallel manipulator has been developed for a wide range of applications such as machine tool applications, motion simulators, and bio-mechanic applications.

In designing a parallel manipulator, Jacobian matrix is usually used. Jacobian matrix is a determinant matrix which is used to solve the inverse of a number of functions with certain variables and used in determining the solution of static analysis, kinematic and dynamic analysis of a parallel manipulator, which is in another words, apart from being used in designing parallel manipulators, the Jacobian matrix method is also used in developing the analogue control system of the aforementioned parallel manipulator. However, this method has its drawbacks because Jacobian matrix can only control a maximum number of 6 outputs, so it can only be used to design a manipulator with no more than 6 actuators. One method to overcome the complexity of the solutions that have been proposed to overcome the limitations of the Jacobian matrix in designing parallel manipulator is by using Inverse Static Analysis (ISA) [1], where one of the existing ISA solution is using artificial intelligence. ISA method is expected to be used to control a parallel manipulator with more than 6 actuators.

In addition to analogue control, there is also discrete control where the actuators are assigned with a limited number of state. A manipulator with discrete control is intended to reduce the complexity of the procedure and to develop a sensorless robot sensors [1, 2]. One example of discrete controlled manipulator is the Binary Snake-like Robot [3-7]. Previous studies which are closely linked to the control of discrete parallel manipulator using artificial intelligence was conducted by Pasila [8, 9]. This study focused on controlling the 6 DOF parallel manipulator using neuro-fuzzy method. The parallel manipulators used in the aforementioned research have 12 prismatic actuator with double-UPU-3D mechanism. UPU means Universal-Prismatic-Universal joints. Actuators used are double action pneumatic actuators that require a number of directional control valves according to the number of actuators. Results obtained from this study is that the parallel manipulator twisted due to the way the actuators are arranged which are separated from each other.

Looking at the current development, there has been less parallel manipulator that own more than 6 actuators that are discretely controlled. Moreover, this kind of manipulator usually use the Jacobian method can produce at maximum 6 outputs. For that reason, a parallel manipulator mechanism with more than 6 actuators needs to be designed according to the needs of artificial intelligence as the ISA solution for the double-hexapod discrete parallel manipulator.

The goal of this paper is to design a parallel manipulator with double-six actuators. The designed double-hexapod manipulator is based on Neuro-Fuzzy control mechanism. The second objective is to obtain a state approximation for each actuator to obtain efficient results with Root Mean Square (RMS) error such as around 5%.

II. RESEARCH METHODOLOGY

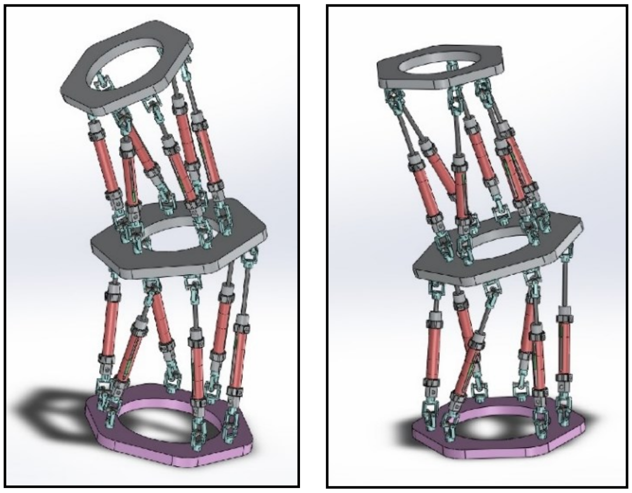
The research methodology in this study is divided into three Section. Section A is about how to design the double-hexapod parallel manipulator, Section B explains how to collect data training, and Section C describes the Adaptive training mechanism using Neuro-Fuzzy architecture.

A. Design of the double-hexapod Parallel Manipulator

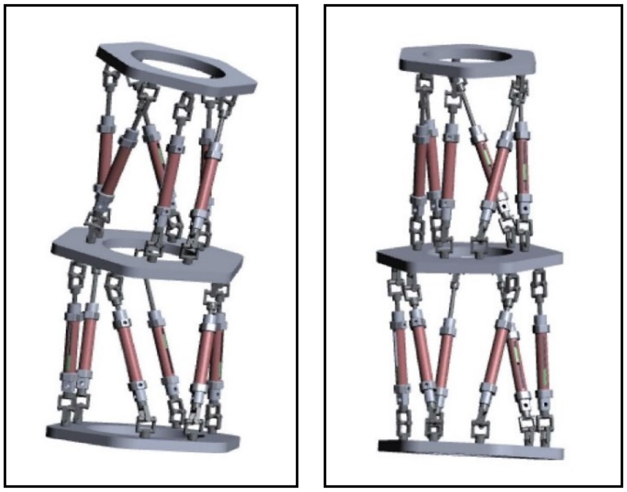
The parallel manipulator design used in this paper consists of three platforms. The medium and the serve as moving platforms and the lower plarform serves as a fixed body, which are connected by 12 pneumatic actuators. All platforms are circular bodies that have different diameters. To determine the dimensions of the fixed body and the moving platform for the manipulator in this research, as well as the location of each actuator, trial and error method is used. Trial and error method was done with the help of simulation software using Solidworks Motion Study. This trial and error method was done in order to obtain the dimensions of the fixed body and the moving platforms to accommodate the actuator arrangement so that the manipulator will not enter the unexpected twist. There are several things that must be considered to determine whether the manipulator will have a twist or not, in this case a parallel manipulator with more than 6 actuator, which are the number of actuators and actuator positions that will affect the dimension of the manipulator. The minimum number of actuators required in order to prevent a twist in the manipulator is 6 actuators, and the maximum number of actuators that can be used is limited only by the dimension specified for the manipulator. In this work, the number of actuator used was determined to be 12 actuators. In order to determine the position of each actuator, the double-hexapod parallel manipulator are designed and tested using Solidworks Motion Study software. From the test results, it was known that the design of the double hexapod discrete manipulator will not twist if it design using the actuator position architecture based on the Stewart-Gough platform [10]. Specifications of all bodies can be seen in Table 1. The design of double-hexapod manipulator can be seen in Fig. 1.

TABLE I
SPECIFICATIONS OF THE DOUBLE-HEXAPOD MANIPULATOR

Material, Mass and Height		
Material	Aluminium 6061	-
Fixed Mass (Buttom)	2477	gr
Medium Mass	3124	Gr
Top Mass	1600	Gr
Minimum Height(end effector)	690	Mm
Maximum Height	860	Mm
Total Mass including joints	10609	gr



(a) (b)



(c) (d)

Fig. 1 End-effector position (a) until (d), rendered CAD view of the double Hexapod Manipulator in several positions.

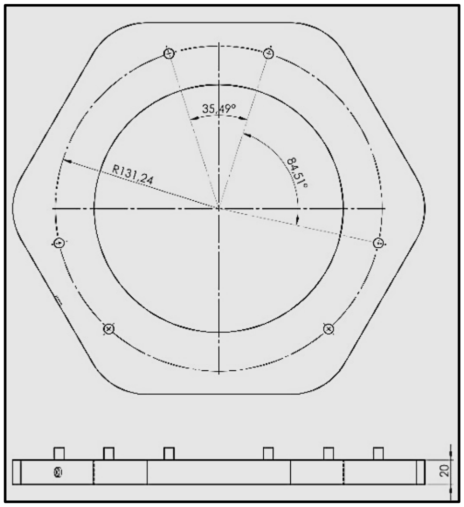


Fig. 2 Fixed Platform of Double Hexapod Manipulator.

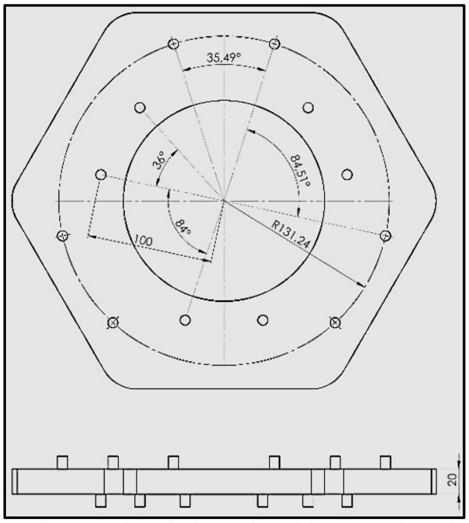


Fig. 3 Medium-Moving Platform of Double Hexapod Manipulator.

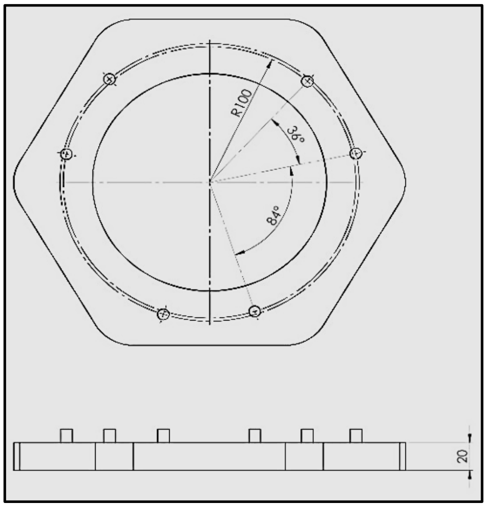


Fig. 4 Top-Moving Platform of Double Hexapod Manipulator.

The both parallel manipulators (medium and top platform) have 2 times 3 pairs of universal joint and giving in total 12 pneumatic actuators which serve as prismatic joints. Actuators connect the moving platform and the fixed body using the universal joints and generate the Universal-Prismatic-Universal (UPU) construction. The UPU construction also applied to the top moving platform. We called the construction joints as double-UPU.

In addition, the valves used are JELPC dual action type pneumatic actuators with 75 mm stroke and can work at air pressure range of 4-8 kg/mm2. Both ends of the actuators are connected to hubs with 25 mm diameter and 21 mm height which are made of ST60 steel. The hubs serve to connect the actuator with the spherical joints. The hub and the spherical joint are then locked by using a pair of plates with a thickness of 20 mm diameter made of ST42 steel.

As a drive system for the manipulator, a compressor, and 12 pieces of 5/3 solenoid valve are used, which are operated by using a Programmable Logic Controller (PLC) Siemens S7-300. Each valve controls a pair of actuators that work in tandem. Actuator is set to work in discrete with 3-states, which can be Retracted, Floated, and Extended.

Moreover, the size of the platforms (bottom, medium and top level) are described in Fig. 2 – Fig. 4.The base platform (bottom) has equal radius with the medium platforms, which the radius has 131.24 mm. The top platform (end effector) has smaller radsius, which is 100 mm. In addition, the twelve-equal actuators are put in the symmetrical position of the related platform.

B. Collecting Data via Solidworks Motion Study

Data gathering was done in two methods, namely software simulation by using Solidworks Motion Study software and manipulator training, which was done by neuro-fuzzy mechanism. Simulation with Motion Study was done by adding 36 contact parameters to prevent solid bodies from penetrating one another. The 36 contact parameters used consist of 12 contacts between the actuator assembly and 24 between the actuator and the corresponding spherical joint. The motion simulation process generates 950 data, where each data consists of coordinates along X, Y, and Z axis of the reference point on the moving platform (end effector). The total generated set of data from simulation (950 data) will be used in artificial intelligence training mechanism that is explained in Section C [9].

C. Adaptive training mechanism using Neuro-Fuzzy architecture

This Section uses Neuro-Fuzzy method as adaptive training mechanism. The architecture is called as feedforward Neuro-Fuzzy Takagi-Sugeno (NFTS), type multi-input multi-output. The inputs of architecture are three set of coordinates XYZ in the related coordinate direction while the outputs consist of the twelve states of the actuators (U_{State}). It also uses Gaussian Membership Functions (GMFs) in the fuzzyfication phase. Moreover, the method is already proved as real-time intelligent controller with very good performance in the term of time settling and rise time [11]. The detailed architecture and explanation of NFTS can be found in [8], where the architecture is Fuzzy structure with learning parameters using LMA.

Moreover, we introduce the GMFs (1) as fuzzyfication functions to the NF methods $G_y{}^n$ (j = 1, 2, 3; n = 1, …, M), for input pairs $\alpha^{\boldsymbol{D}} = [P^X, P^Y, P^Z]$, where α^D are the input set of the positions (P^X, P^Y, P^Z) of the moving platform with respect to the XYZ Euler coordinates.

$$G_y^n\left(\alpha_y\right) = exp\left(-\left[\left(\alpha_y - c_y^n\right)/\sigma_y^n\right]^2\right) \qquad (1)$$

with the Gaussian's parameters G_y^n have means c_y^n and variance σ_y^n together with the corresponding n-fuzzy rules (FR^n) can be written as:

$$\begin{aligned} &FR^n : IF\ \alpha_1 is\ G_1^n\ AND \alpha_2 is\ G_2^n\ \ AND\ldots \\ &THEN\, y_i^n = w_{0i}^n + w_{1i}^n \alpha_1 + \ldots \end{aligned} \qquad (2)$$

Here w_{0i}^n, w_{1i}^n being the Takagi-Sugeno weights (for i = 1, …, 3, and n = 1, …, M, M is the number of optimized rules for the proposed model, M = 5 will be found by optimized search mechanism), the last part of the considered Neuro-Fuzzy model calculates the output variables $\bar{U}$.

$$\bar{U} = \sum_{n=1}^{M} y_i^n \left(\frac{\prod_{y=1}^{12} G_y^n(\alpha_y)}{\sum_{n=1}^{M} \prod_{y=1}^{12} G_y^n(\alpha_y)}\right) \qquad (3)$$

The NF output (3) consits of twelve actuator states (outputs) and still in the real form. Moreover, the outputs will be derived by alternatively approximating the activation states of actuators U_i through the following threshold function:

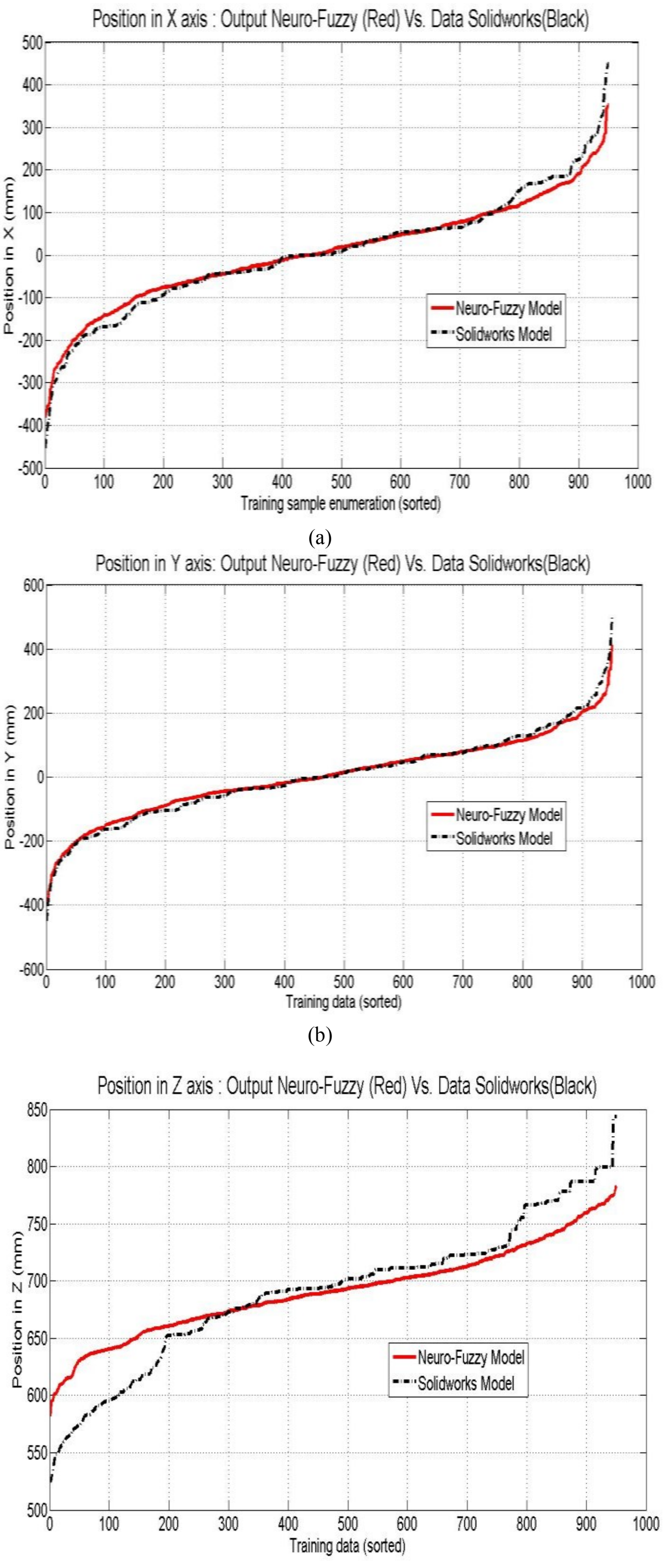
$$U_i = round\left(\overline{U}_i\right) \tag{4}$$

where *round* indicates a process to change real form of $\overline{U}$ into the three state numbers which are -1, 0 and +1. The equation (4) can be processed because the round convert all analogue output value from -1, 0 to +1 into the nearest three values above. Moreover, in the process to find the predicted outputs, NFTS model requires the tuning of the parameters c_y^n, σ_y^n, w_{0i}^n, w_{yi}^n (here y = 1, 2, 3; i = 1, 2,...,12; n = 1,…, 5;). The number of parameters for the considered architecture (with M=5) is 285 parameters from four tuned parameters. The values of these parameters are found by an optimized learning procedure with around one million times of iteration. The learning procedure employs 100% of the input-output correspondences known from Δ dataset for the double-six-three states respectively.

III. Results and Discussions

In this paper, we generate the data simulation from the 3D simulation software. Here, Fig. 5 shows graphs of data simulation results with a total of 950 data which are already sorted from the smallest to the largest value. The results of the collected data are obtained from Solidworks Motion Study software.

The simulation results show that the position of aforementioned reference points along the X, Y and Z axis have similar results compared to their neuro-fuzzy results. The parallel manipulator is planned to be controlled discretely using Neuro-Fuzzy as ISA solution for the double-hexapod manipulator. The performance of the discretely controlled manipulator is expected to resemble the analogue controlled manipulator. Fig. 5 describes the comparison between the simulation results obtained with the Solid works Motion Study software, which shows the approximate value when the actuator is controlled discretely, and position when approached using analogue control. In addition, it can be seen that the position along X, Y and Z axis closely have generated similar value compared to the continuous controller form. On the other hand, there is a fairly large deviation between the coordinates generated from the simulation with the software and the coordinates generated when using the trained results. It can be inferred that simulation along the Z axis has biggest error compare to Axis X and Y at some point. As a result, the total performance has, in average, 3.39% of RMSE (root mean squared error).



(a)

(b)

(c)

Fig. 5. Data Graph Showing Comparison between Software Simulation Result and Training Results from Neuro-Fuzzy Method: (a) Position along the X axis (b) Position along the Y axis (c) Position along the Z axis

The trained data from neuro-fuzzy method should be compared with the software simulation data to obtain model error which is expressed as root mean square error (RMSE). The RMSE is calculated for each displacement error, and it is expressed in mm and percent error. The result of RMSE of each axis can be seen in Table 2. Moreover, data comparison samples between the position obtained by simulation using the Solidworks Motion Study software and position measurement results of NFTS network also can be seen in Table 3.

TABLE II
COMPARISON OF RMSE BETWEEN THE RESULTS OF NEURO-FUZZY NETWORK AND MANIPULATOR SIMULATION USING SOLIDWORKS

RMSE	Error Percentage
X Coordinate	2.16%
Y Coordinate	2.06%
Z Coordinate	5.97%
Average RMSE	3.39%

TABLE III
COMPARISON OF MANIPULATOR POSITION RESULTS BETWEEN SOFTWARE SIMULATION RESULT AND MANIPULATOR TRAINING FROM NEURO-FUZZY NETWORK (12 SAMPLES)

No	Simulation Final Coordinates (mm)			NFTS Performances Coordinates (mm)		
	X	Y	Z	X	Y	Z
1	-452	-447	525	-381	-392	582
2	-434	-417	526	-369	-393	586
3	-431	-417	526	-368	-392	595
4	-414	-363	532	-357	-355	595
5	-396	-360	533	-355	-355	596
6	-393	-360	534	-355	-340	596
7	-377	-349	540	-350	-335	600
8	-452	-447	525	-381	-393	583
9	432	386	843	345	334	775
10	435	419	843	345	334	777
11	449	456	843	351	375	782
12	449	506	843	351	412	782

From the Table 3, it can be seen that some data samples on the X,Y and Z axis, will generate greater error than its average error. This is due to the construction of the manipulator mechanism which was still not the optimized version and only can be reduced by design. Moreover, from the error calculation in Table 2, it can be seen that the mechanism RMSE is 2.16% along the X axis, 2.06% along the Y axis, and 5.97% along the Z axis, while the RMSE observed along the X, Y, and Z axis combined is 3.39%. Therefore, it can be said that although the construction of the manipulator is still not yet optimized, the double-hexapod parallel manipulator mechanism designed in this research is working well with small error (below 5%).

IV. CONCLUSIONS

In conclusion, this paper discusses about the design and modeling of a double-hexapod parallel manipulator which is used for the implementation of NFTS network as the control strategy of double-hexapod parallel manipulator. The simulation result obtained using Solidworks Motion Study shows that the reference point on the moving platform can move along the X, Y, and Z axis, indicated by the position of the point along each axis, which are between -452 mm and 449 mm along the X axis, between -447 mm and 506 mm along the Y axis, and between 525 mm and 843mm along the Z axis respectively. The magnitude of position generated using Solidworks Motion Study will be used as inputs in the network learning process. RMSE of the manipulator obtained by comparing software simulation results and the neuro-fuzzy training data shows relatively small error values on the X axis and Y axis data, but bigger error in Z axis.

The conclusion that can be drawn from this research based from the value of the RMSE is that the parallel manipulator double-hexapod actuators are working (average error RMSE around 3.39%) and can use artificial intelligence method as control strategy for ISA solution.

ACKNOWLEDGEMENT

The authors would like to thank Petra Christian University, Surabaya and Department of Higher Education of Indonesia (Menristek-Dikti) for supporting this research under two years Research Grant 2014-2015, with the Number: 25/SP2H/LPPM-UKP/IV/2014.

REFERENCES

[1] Pieper D.L., “The Kinematics of Manipulators under Computer Control”, PhD Thesis, Stanford University, Stanford, CA, 1968.

[2] Roth B., Rastegar J. and Sheinman V., “On the Design of Computer Controlled Manipulators”, First CISM-IFTMM Symposium on Theory and Practice of Robots and Manipulators, pp. 93-113, 1973.

[3] Maeda K., Konaka E.,” Novel Inverse Kinematics Solution Algorithm for Binary Manipulator Using Ellipsoidal Outer-Approximation of Workspace”, SICE Journal of Control, Measurement, and System Integration, Vol. 8, No. 1, pp. 044–051, 2015.

[4] Chirikjian G. S.,”Inverse Kinematics of Binary Manipulators Using a Continuum Model”, Journal of Intelligent and Robotic Systems, vol.19, pp.5-22, 1997.

[5] Ebert-Uphoff I., Chirikjian G.S., “Inverse Kinematics of Discretely Actuated Hyper-Redundant Manipulators Using Workspace Densities”, Proceedings of the 1996 IEEE International Conference on Robotics and Automation, pp. 139-145, 1996.

[6] Suthakorn J. and Chirikjian G. S.,”A New Inverse Kinematic Algorithm for Binary Manipulators with Many Actuators”, Advanced Robotics, vol. 15, n. 2, pp. 225-244, 2001.

[7] Sujan V.A., Lichter D., Dubowsky S., “Lightweight Hyper-Redundant Binary Elements for planetary Exploration Robots”, 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1273-1278, 2001.

[8] Pasila F., “Inverse Static Analysis of Massive Parallel Arrays of Three-State Actuators via Artificial Intelligence”, PhD Dissertation, University of Bologna, 2013.

[9] Pasila F., Alimin R., Natalius H., “Neuro-Fuzzy Architechture of the 3D Model of Massive Parallel Actuators”, ARPN Journal of Engineering and Applied Sciences, vol. 9, pp. 2900-2905, 2014.

[10] Ioannis D., Papadopoulos E., “Model-based control of a 6-dof electrohydraulic Stewart–Gough platform”, Mechanism and Machine Theory 43, pp.1385–1400, Elsevier, 2008.

[11] Pelusi D., “Designing neural networks to improve timing performances of intelligent controllers”, Journal of Discrete Mathematical Sciences and Cryptography, 16 (2-3), pp. 187-193, 2013