

# ICAICAT 2016

*by Roche Alimin*

---

**Submission date:** 29-May-2022 09:14PM (UTC+0700)

**Submission ID:** 1846414535

**File name:** 2016-10-43-0082\_-\_Felix\_Pasila-Roche\_Alimin.pdf (1.4M)

**Word count:** 3819

**Character count:** 19607

# Applications of Artificial Intelligence Control for Parallel Discrete-Manipulators

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

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

**Abstract**—Parallel Discrete-Manipulators are a special kind of force regulated manipulators which can undergo continuous motions despite being commanded through a large but finite number of states only. Real-time control of such systems requires very fast and efficient methods for solving their inverse static analysis. In this paper, artificial intelligence techniques (AI) are investigated for addressing the inverse static analysis of a planar parallel array featuring ten three-state force actuators and two applications using 3D Massively Parallel Robots (MPRs) with one and two layers. In particular, the research method used simulation software and hardware testing with the case of parallel manipulator with two level discrete pneumatic actuators. Simulations with typical desired displacement inputs are presented and a good performance of the results compared to AI is obtained. The comparison showed that the parallel manipulator has the Root Mean Squared Error (RMSE) has less than 10% and can be used for controlling the ternary states of discrete manipulators via AI.

**Keywords**—discrete-manipulators; artificial intelligence; inverse static analysis; three-state actuators

## I. INTRODUCTION

The ways discrete-manipulators work are analogue to the muscle fibers work. In general, one muscle fiber, called as muscle cell, consists of hundreds single fiber and each single fiber has filaments and they are constructed in the arrays fashion (the cells and filaments are constructed in serial and parallel arrays). The cells produce mechanical force for contraction when the neuron of motor unit (one motor unit communicates with some muscle fibers) releases stimulus/trigger to some related fibers with constant value. We can say that the simple mechanism of single muscle is essentially analogue to the discrete-manipulators (DMs) that constructed in arrays (in serial and/or parallel).

Other motivation is that DMs with 3-states have been used in different applications of robotics and biomechanics. These are discrete devices, in which the states flip between a finite numbers of possible values. In particular we consider actuators with three stable positions: the positively, the neutral and the negatively ones (in muscle contraction they called these states as: concentric, isometric and eccentric). Possible designs involve for instance solenoids, pneumatics, dielectric elastomer actuators, shape memory alloy. Because of the simplicity of

their design, these kinds of actuators have many potential benefits: they are relatively cheap and lightweight. In contrast with the advantages, the main limitations to the use of such architectures come from the complexity of their activation control, which results to be nonlinear, discontinues, and the number of achievable configurations also being exponentially proportional to the number of actuators. The motivations above lead us to define the main goals: to develop a general predictive control for activating 3-state actuator arrays with real time control (fast response); and to apply the control mechanism for several actuators design.

The DMs are a special kind of mechanisms whose actuators can only be made switching among a finite number of states. Introduced first in the 1970's with the challenge to consider sensor-less robots as well as to reduce the complexity of control mechanism [1]. Recently, DMs can be classified into two groups depending on whether their actuators act as either discrete displacement generators or discrete force generators. Examples of DMs of the first type are the binary Snake-Like Robots (SLR) [2-4], which are kinematically constrained mechanisms employing a large number of bi-stable actuators whose configuration can be either fully contracted (inactivated state) or fully extended (activated state) irrespective of the arbitrary external forces/moments on them. Examples of DMs of the second type are the binary Massively Parallel Robots (MPR) [5,6], which are dynamically constrained mechanisms employing a large number of on-off actuators that exert either a constant force (active state) or no force (inactive state) irrespective of their arbitrary kinematically unconstrained configuration.

Moreover, owing to the large number and the discrete nature of the actuator variables (positions for SLR and for MPR), the inverse kinematic analysis of SLR and the inverse static analysis of MPR are usually very difficult problems whose solution practically requires complicated techniques. In the past, significant research efforts have been devoted to address these inverse problems, in particular by resorting to: exhaustive brute-force search approaches, methods of differential geometry and calculus, combinatorial algorithms, search algorithms using genetic computation, probability method, and implemented Hopfield Networks [2-6]. Despite most of the proposed solution schemes are formally very elegant and rather effective in reducing problem

complexity from exponential time to polynomial time, the resulting algorithms still require too many calculations for real-time manipulator control.

In this framework, the present paper showing some applications of artificial intelligence methods for the real-time solution of the Inverse Static Analysis (ISA) of 1 DOF-MPR (planar) and 3D-MPR actuated by 10 or more three-state force generators. In particular Neuro-Fuzzy and Recurrent Neural Networks models for the ISA of such a ternary MPR are first constructed and then their real-time computation performances are compared to the best preparation time  $t_p$ , online computing time  $t_c$ , and the best generalization error  $e_g$  respectively.

## II. INVERSE STATIC ANALYSIS SOLUTION VIA ARTIFICIAL INTELLIGENCE METHODS

### A. Inverse Static Analysis Model

In this paper, we explore the possibilities of using artificial intelligence (AI) methods for the real-time solution of the Inverse Static Analysis (ISA). The ISA problem in this context extends to find the best combination of the activation states

$u_i^*$  (among a total of  $3^m$  possibilities for any desired position angle  $\alpha^D$ ) which enables the generation of the moment  $M^*$  (namely  $M^* = M(\alpha^D, u_i^*)$ ) that more closely matches a desired torque  $M^D$ ; that is, to find the state combination  $u_i^*$ ,  $i = 1, \dots, m$ , and  $m$  = number of actuator arrays for which the error  $e^* = \|M^D - M^*\|$  is

$$e^* = \min_{u_i \in \{1,0,-1\}} \|M^D - M(\alpha^D, u_i)\| \quad (1)$$

Notice that since the desired  $M^D$  can be any real value, whereas the range  $M$  is only a discrete subset, in general the minimum error  $e^*$  is different from zero. Moreover, owing to the discrete nature of the  $m$  variables  $u_i$ , the ISA described by Eq. (1) cannot be solved via standard pseudo-inverse equations.

### B. Artificial Intelligence learning methods for ISA solution

We introduce the Lavenberg-Marquardt Algorithm (LMA) as a learning mechanism that is used in AI methods for ISA solution. The LMA equation according to [7] can be written as a function  $V(w)$  is meant to minimize with respect to the all LMA parameter  $w$  using Newton's method. The update of parameter vector  $w$  is defined as updated equation below:

$$w(n+1) = w(n) - [J^T(w) \cdot J(w) + \alpha \cdot I]^{-1} \cdot J^T(w) \cdot e_r(w) \quad (2)$$

where  $n$  is constant value: 1,2,3 ...,  $J(w)$  is Jacobian matrix,  $\alpha$  is a constant value which is multiplied or divided by some factor whenever the iteration steps increase or decrease the value of  $V(w)$ .

The LMA can generated the training data from the 3D SW software or kinematic equations. At this point,  $m$  number of actuators will produce  $3^m$  training dataset (at least three information), which are: forces/moments, position and the related three-state combination respectively. In the ISA model, the input data will be forces and/or position and the output is the state combination related to the inputs. In the learning process, the LMA try to find the optimum parameters by minimizing the  $V(w)$  via Eq. (2).

## III. EXAMPLE OF THREE-STATE CONTROL MECHANISM

### A. CSL-3RP Planar Mechanisms

The ternary Planar Mechanisms considered in this Section is represented in Fig. 1. It consists of ten same Crank and Slotted-Lever (CSL) 3RP planar mechanisms ( $R$  representing revolute joint and  $P$  representing prismatic joint) sharing the same moving revolute joint. The common crank is hinged through a  $R$  joint to the fixed frame at point  $O$ , the ten sliders of the different  $P$  joints are hinged by a common  $R$  joint to the crank at a common moving point  $A(\alpha)$  ( $\alpha$  indicating the angular position of the crank with respect to a fixed frame), the ten linear guides of the different  $P$  joints) are hinged to the fixed frame via different  $R$  joints at the points  $B_i$  that are equally spaced along a circular curve with angular span equaling  $162^\circ$  with radius  $r$ . For more explanation about the performance of compared ISA methods and its testing performance can be seen in Table 1 and Fig. 2 respectively [7].

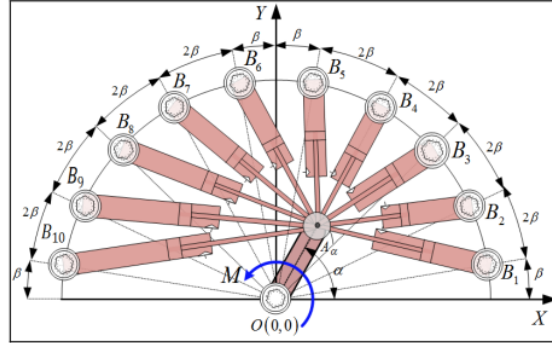


Fig. 1. Ternary Massively Parallel Robot (MPR) actuated by ten three-state force generators

Table 1 shows the comparison of several ISA methods concerning preparation and computation time and their generalization error. The performance demonstrate that HN and NFTS are the suitable method concerning the best preparation time  $t_p$  and online computing time  $t_c$  respectively, and the HN method has the best generalization error  $e_g$ . In contrast, concerning minimum requirements of ISA solution such as real-time computing and generalization error, the ERNN and MLP methods are the recommended procedures dealing with ISA problem [7].

TABLE I. PERFORMANCE COMPARISON OF THE 10-TERNARY MPRS OF THE CONSIDERED METHODS TABLE STYLES

Method <sup>a</sup>	NFTS	NFLUT	MLP	ERNN	HN
$t_p$ (s)	7.1e3	7.3e3	3.1e4	7.8e3	892
$t_c$ (s)	1.9e-3	3.8e-2	3.3e-3	3.3e-3	0.20
$e_m$ (N)	0.622	0	0.346	0.300	0
$e_g$ (N)	0.998	0.3966	0.379	0.335	0.30
RMSE (%)	19.9	7.9	7.6	6.7	6.01

<sup>a</sup> The CPU has 32bit OS, dual core, 2.6 GHz. NFTS(Neuro-Fuzzy Takagi Sugeno), NFLUT(NFTS with Look-Up Table), MLP(Multilayer Perceptron Neural Network), ERNN(Elman Recurrent Neural Network), HN(Hopfield Network)

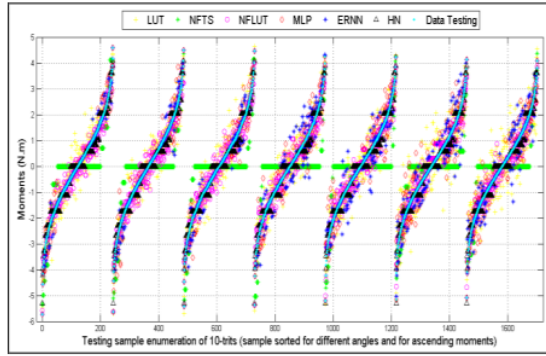


Fig. 2. Testing Performance of 10-ternary MPR with Different ISA Methods

### B. The designing of the 3D parallel manipulator with 16 discrete-actuators

The parallel manipulator design used in this part consists of a pair of body: the upper body that serves as a moving platform and the lower body serves as a fixed body, which are connected by 16 discrete-pneumatic actuators. Both the upper body and the lower body are circular forms that have altered diameters. The experimental method with discrete combination is used to determine the dimensions of the fixed body and the moving platform for the manipulator, as well as the location of each actuator. In this case, the discrete combination method was done with the help of simulation software using Solidworks. This combination method was done by: 1) determining the dimension of the fixed body and the moving platform to accommodate the actuator arrangement so that the manipulator will not experience the unexpected twist and 2) collecting the minimum combination states and positions that gives stable positions [8].

There are several things that must be considered to determine whether the manipulator will experience a twist or not, in this case a parallel manipulator with more than six

actuators, 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 six pneumatic actuators, and the maximum number of actuators that can be used is limited only by the dimension specified for the robot manipulator. In this paper, the experimental number of actuator used was determined by choosing sixteen actuators.

Moreover, the proposed architecture should have specifications of both bodies, like shown in Table 1 and the proposed manipulator can be seen also in Fig. 3.

TABLE II. SPECIFICATIONS OF THE ROBOT MANIPULATOR MOVING PLATFORM

Moving platform		
Material	Aluminium 6061	-
Mass	6758.56	gr
Volume	2503170.76	mm <sup>3</sup>
Outer Circle Diameter	560	mm
Diameter of Centre Joint	500	mm
Inner Circle Diameter	400	mm

The parallel manipulator used has sixteen pairs of spherical joint and 16 pneumatic actuators which serve as prismatic joints. Actuators connect the moving platform and the fixed body using the spherical joints to form Spherical-Prismatic-Spherical (SPS) construction. Actuators used are JELPC dual action type pneumatic actuators with 70 mm stroke and 12 mm bore and can work well at air pressure range of 4-9 kg/mm<sup>2</sup>. Both ends of the actuators are connected to the 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 1 mm 30 mm diameter made of ST42 steel.



Fig. 3. Robot manipulator using 16 discrete actuators.

The simulation results show that for the position of the aforementioned reference point along the X-axis, the maximum value is 64.63 mm and the minimum value is -64.64 mm, along the Y-axis the maximum value is 276.14 mm and the minimum value is 199.42 mm, and along the Z-axis the maximum value is 64.62 mm and the minimum value is -64.67mm. Along the X axis, the maximum force is 450.64 N, and the minimum force is -450.64 N, On the Y axis, the



maximum force is 2154.78 N, and the minimum force is -2154.78 N, and along the Z axis, the maximum force is 450.64 N and the minimum force is -450.64 N. The graphs for both coordinates and force along the Y axis look different from other graphs due to the data value not being evenly distributed.

The parallel manipulator is planned to be controlled discretely using Neural Network as ISA solution for the manipulator. The performance of the discretely controlled manipulator is expected to resemble the analogue controlled manipulator. From Fig. 4, we can see 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 and force when approached using analogue control. In addition, it can be seen that the position and the force along X and Z axis closely resemble the value generated when using analogue controller. On the other hand, there is a fairly large deviation between the coordinates and the force generated from the simulation with the software and the coordinates and the force generated when using the analogue control observed along the Y axis which can be seen in the graph, where the position and force results obtained using the simulation along the Y axis jump at some point. As a result, it is possible that neural network might not work optimally as an ISA solution for the planned manipulator.

Fig. 4 shows a comparison chart between 105 data that has been selected from the simulation and measurement data that has been sorted from the smallest to the largest value.

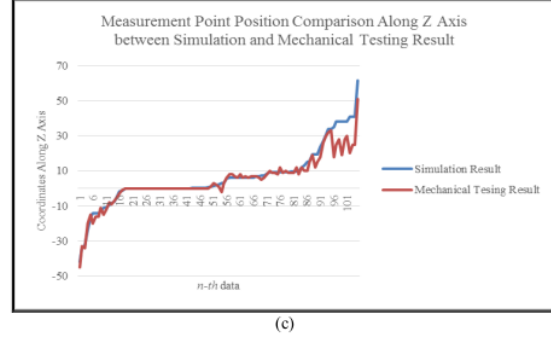
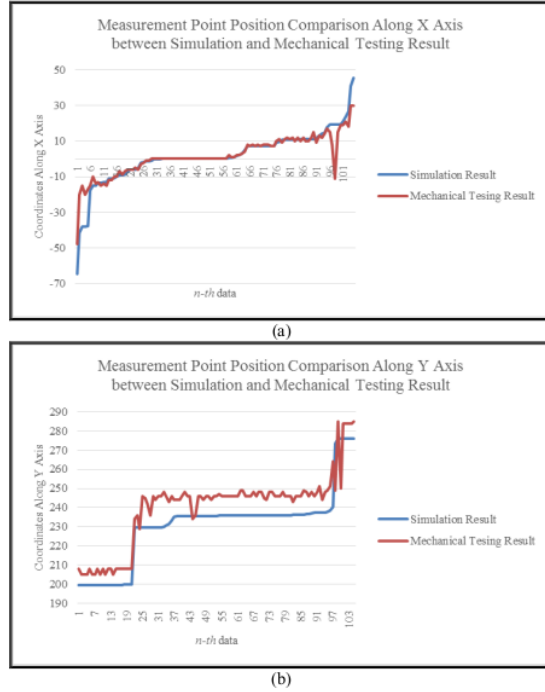


Fig.4. Data Graph Showing Comparison between Software Simulation Result and Manipulator Measurement Process Result (a) Position along the X axis (b) Position along the Y axis (c) Position along the Z axis

The mechanical test data needs to be compared with the software simulation data to obtain mechanism error which is expressed as root mean square error (RMSE). RMSE obtained, expressed in mm and percent error, can be seen in Table 3, while some data comparison samples between the position obtained by simulation using the Solidworks Motion Study software and position measurement results obtained by manipulator prototype testing can be seen in Table 3.

TABLE III. RMSE OBTAINED BY COMPARING THE RESULTS OF MECHANICAL TESTING AND THE RESULTS OF MANIPULATOR SIMULATION USING SOFTWARE

RMSE	mm	Error Percentage
<i>X Axis</i>	0.57692	5.872%
<i>Y Axis</i>	1.0598	0.451%
<i>Z Axis</i>	0.47052	6.053%
<i>Average RMSE</i>	0.43171	2.815%

### C. Two layers Hexapod 3D-MPR

The parallel manipulator design used in this paper consists of a pair of body, the upper body that serves as a moving platform and the lower body that serves as a fixed body, which are connected by 12 pneumatic actuators. Both the upper body and the lower body 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 (SW). This trial and error method was done to obtain dimensions of the fixed body and the moving platform to accommodate the actuator arrangement so that the manipulator will not experience an unexpected twist. 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 research, the number of actuator used was determined to be 12 actuators. In order to determine the position of each actuator, a novel parallel manipulator was design which based on hexapod Stewart-Gough platform [9, 10].

TABLE IV. SELECTED DATA OF TWO-SIX HEXAPOD MECHANISM  
(1 = EXTEND, -1=RETRACT AND 0 = FLOATING)

Lower Manipulator States						Upper Manipulator States						Axis Coordinates		
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	X	Y	Z
0	0	1	1	0	0	1	0	-1	-1	0	1	6	24	627
0	0	1	1	0	0	1	1	0	-1	-1	0	6	27	623
0	0	0	1	1	0	1	0	-1	-1	0	1	-85	-83	574
0	0	0	0	1	1	0	1	1	0	-1	-1	-18	-4	628
0	1	1	0	-1	-1	0	1	0	1	0	0	50	15	707
-1	-1	0	1	1	0	1	0	-1	-1	0	1	-9	19	623
-1	-1	0	1	1	0	-1	0	1	1	0	-1	9	51	574
1	1	0	-1	-1	0	1	0	-1	-1	0	1	10	-22	627
-1	0	1	1	0	-1	1	1	0	-1	-1	0	-11	-12	619
0	-1	-1	0	1	1	1	0	0	0	0	0	15	-2	728

The motion simulation process generates 1596 data, where each the data consists of coordinates along X, Y, and Z axis of the reference point on the moving platform, and the total force on the X, Y and Z axis. The measurement of position of the aforementioned point on the moving platform is done with the help of a needle and light to highlight the position along X and Z axis of the reference point on the moving platform. Some of the extracted data can be seen in Table 4.

In this Section, the data simulation is generated from the 3D SW software. At this point, Fig. 5 and 6 show the implementation of the discrete manipulator with 12 actuators along with the graphs of data simulation results and their neuro-fuzzy model respectively. The total dataset for model use 1596 data which are already selected and sorted from the smallest to the largest value.

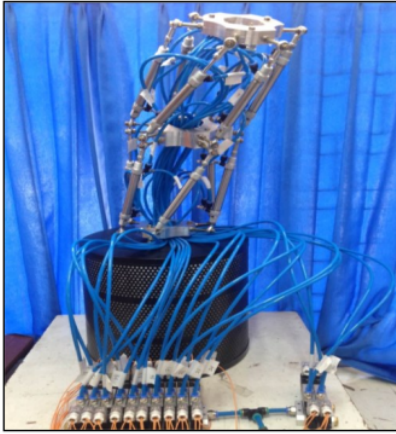
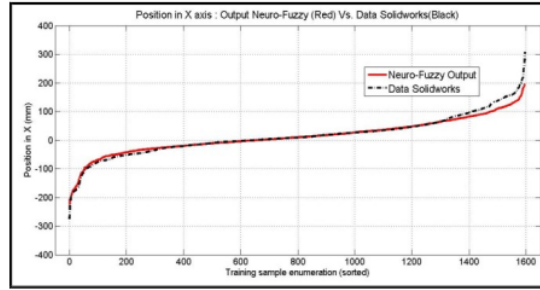


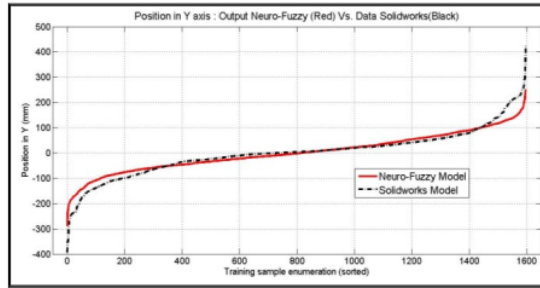
Fig. 5. Implementation of Discrete Manipulator with 12 Discrete Actuators

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 two-six manipulator. Moreover, Fig. 6 describes the comparison between the simulation results obtained with the SW software, which shows the approximate value when the actuator is discretely controlled. 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. As a result, the total performance has, in average, 2.12% of RMS error.



(a)



(b)

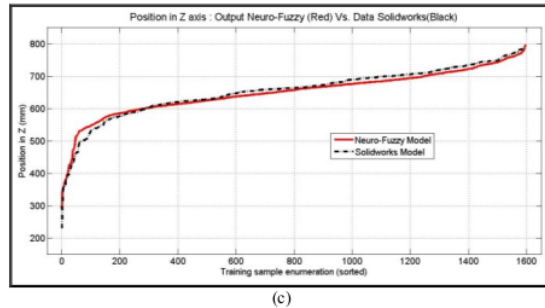


Fig. 6. Data Graph Showing Comparison between Software Simulation Result and Manipulator Measurement Process Result (a) Position along the X axis (b) Y axis (c) Z axis

#### IV. RESULTS AND DISCUSSIONS

As conclusion, this paper presented: 1) one planar massively parallel robots (MPRs) with 10 three-state force actuators and one continuous degree of rotational motion and its ISA Solution using two Neuro-Fuzzy methods (NFTS, NFLUT) and three Neural Network methods (MLP, ERNN and HN) of the considered MPRs. Thanks to the partitioned and spatially distributed actuator architecture, the considered discrete robot features rather sufficient, identical and accurate torque generation capabilities, compared to the standard CSL mechanism (actuated by a single continuously regulated force generator); 2) two 3D-MPRs (sixteen actuators and two-six actuators) with design and mechanical testing using Neuro-Fuzzy method. Therefore it is most likely that according to [7], neuro-fuzzy and neural network can be used as ISA solution on this discrete robot manipulator. The conclusion that can be drawn from this research based from the value of the RMSE is that the parallel manipulator sixteen actuators are designed in this research works relatively well with mechanical testing error RMSE below 10%.

#### ACKNOWLEDGMENT

The first author would like to sincerely thank his advisers, Prof. Parenti-Castelli, Dr. Veretichy and Dr. Berselli, for their immense help, encouragement and support through this research work. Also the authors would like to thank Department of Higher Education of Indonesia for supporting this research under three years Research Grant 2016-2018, with the Number: 1/SP2H/PDSTRL\_PEN/LPPM-UKP/V/2016.

#### REFERENCES

- [1] 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.
- [2] Chirikjian G. S., "Inverse Kinematics of Binary Manipulators Using a Continuum Model", *Journal of Intelligent and Robotic Systems*, vol.19, pp.5-22, 1997.
- [3] 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.
- [4] 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.
- [5] Lichter D., Sujan V.A., Dubowsky S., "Computational Issues in the Planning and Kinematics of Binary Robots", *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp. 341-346, 2002.
- [6] Yang P., Waldron K.J., "Massively Parallel Actuation", 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 868-873, 2001.
- [7] Pasila F., "Inverse Static Analysis of Massive Parallel Arrays of Three-State Actuators via Artificial Intelligence", PhD Dissertation, University of Bologna, 2013.
- [8] Pasila F., Alimin R., Natalius H., "Design of Massive Actuators for 3D Robot manipulators", *Proc. of International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2015)*, Palembang, Indonesia, 19 -20 August 2015
- [9] 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
- [10] Pasila F., Alimin R., "Neuro-Fuzzy Architecture of the 3D model of Massive Parallel Actuators", *ARPN Journal of Engineering and Applied Sciences*, Vol 9/No. 12/2014

# ICAICAT 2016

---

## ORIGINALITY REPORT

---

9%

SIMILARITY INDEX

9%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

---

## MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

---

9%

★ **ajbasweb.com**

Internet Source

---

Exclude quotes      On

Exclude bibliography      On

Exclude matches      Off