Proceedings of the 1997 IEEE
International Conference on Robotics and Automation
Albuquerque, New Mexico - April 1997

Task Based Design of Modular Robot Manipulator using Efficient Genetic Algorithm

Jeongheon Han* W. K. Chung† Y. Youm‡ and S. H. Kim§

Robotics Lab., School of Mechanical Engineering
Pohang University of Science & Technology(POSTECH)
Tel:+82-562-279-2844; Fax: +82-562-279-5899
E-mail: jhhan@risbot.rist.re.kr

Advanced Robotics Development
Korea Atomic Energy Research Institute
150 Dukjin-Dong, Yusung-gu, Taejon, Korea

Abstract

Modular robot manipulator is a robotic system assembled from discrete joints and links into one of many possible manipulator configurations. This paper describes a task based design method of modular robot manipulators. A locking mechanism which provides quick coupling and decoupling is developed and a parallel connection method is devised reducing the number of components on each module.

To automatically determine the optimal link lengths of modular manipulator for a given task, the algorithm is two step: determine the necessary configuration of robot using kinematic equations and then determine the optimal link length using the proposed efficient genetic algorithm. Some of design examples are shown.

1. Introduction

Till now, many robot manipulators have been developed for practical use. However, most of them cannot change their structure, so it is difficult to adapt such a system to variable tasks and environments. Easily reconfigurable robotic system has long been desired to accomplish such tasks. The term modular robot manipulator is referred to a robot manipulator assembled from discrete mechanical joints and links into one of many possible manipulator structures [8]. It is composed of interchangeable links and joint modules of various sizes. By recombining the modules, different robot can be created so as to meet a variety of task requirements using standard mechanical and electrical interfaces. Such a manipulator has several advantages over conventional manipulators; low cost, easy maintenance, easy modification and durability against system malfunctions. The feasibility of the modular robot manipulator has been carried out in the prototype systems built in several research institutes [1, 7, 8, 13], and kinematic analysis of modular robot manipulator has also been performed in [6, 10].

Effective use of the modular manipulator requires task based design software. The software generates optimal available configuration according to the input descriptions of the task. Several different approaches of module assembly planning and task optimal configurations were shown in [3, 14, 15].

The objective of this paper in the area of mechanical design of modular manipulator is to develop such a software, which will allow a user to configure suitable robot geometry for a set of given tasks.

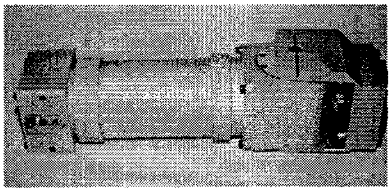
*Graduate Student in Ph.D Program, POSTECH
†Associate Professor, POSTECH & ARC
‡Professor, POSTECH
§Senior Research Engineer, KAERI

2. Mechanical Design

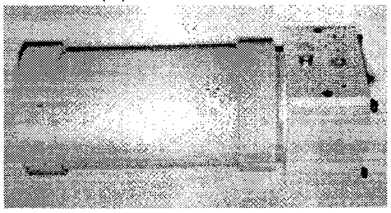
The developed manipulator is composed of three kinds of modules: manipulator base, link module and two pivot joint modules. The design of each module is independent of other modules except for standardized module interfaces. The base module and link module have no degrees of freedom.

2.1. Joint Module

For modular robots, the actuator has to be located in the joint module. Generally three types of 1-DOF joint modules can be considered; revolute(rotating and pivoting) and prismatic joints. A rotating type joint has a link axis which is co-linear with each other to the joint axis. A pivoting one has link axis which is perpendicular to the joint axis. For a prototype of modular manipulator, we built only two pivoting type joint modules. Our prototype design for the pivot joint is shown in Fig. 1(a). It compactly contains a DC motor with encoder and gear transmission(the reduction ratio is 1:100) and is rated at regular speed of 30[rpm] and regular torque of 30[Nm]. Mass of each joint module is approximately 6.5[kg]. There are a pair of locking element at both sides of joint module.



(a) Joint Module



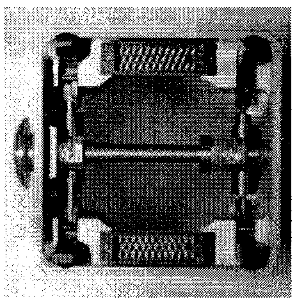
(b) Link Module

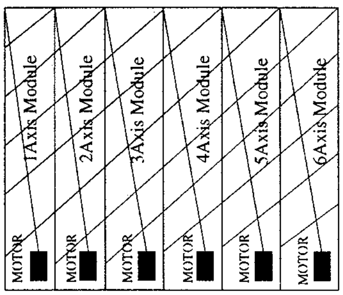
Fig. 1 Basic Module

2.2. Link Module

The purpose of the link module is to change the distance between the rotational axis of the adjacent joint modules. The link module is shown in Fig 1(b). It consists of two cylinders, one inserted into the other. Two cylinders can be linearly translated manually relative to each other in the direction of the link, and can be locked by adjusting ball into the hole. There is no offset and twisting between them.

2.3. Locking Mechanism

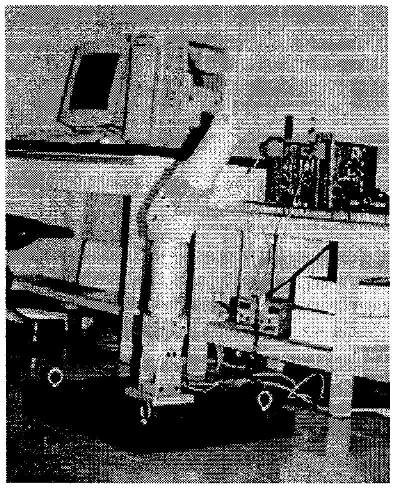




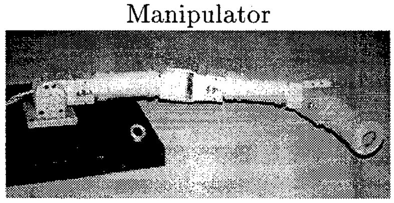
(a) Locking Mechanism (b) Electrical Parallel Connection

Fig. 2 Standardized Connector

One of the major difference between modular manipulator and conventional manipulator is the standardized interace component. In order to assemble the joint and link modules, locking mechanism is required. The connection must both align the modules and lock them together with sufficient strength. We developed a quick-coupling mechanism shown in Fig. 2(a) with which a secure mechanical connection between modules can be achieved by simply plugging by hand; no tools are required. The mechanical coupling is accomplished using kinematic linkage and spring. Inside the locking mechanism are two pair of modular connectors which have 34- electrical pins on both sides. These correspond to matching female components on the mating connector. Sets of pins are wired in parallel to carry power and control signals. A parallel connection method is developed for better communication and the principle of it is shown in Fig. 2(b). As shown in this figure, electrical connection of each module is the same, so any connection of module ahs same result. Owing to the parallel connection method, control input signal and sensor feedback signal are directly transferred to the control system. Another important advantage of the parallel connection is that the number of components on each module is greatly reduced and it becomes easy to control and achieves small and lightweight module, which is extremely effective for improving the payload capacity and tip speed.



(a) One D.O.F Modular Manipulator



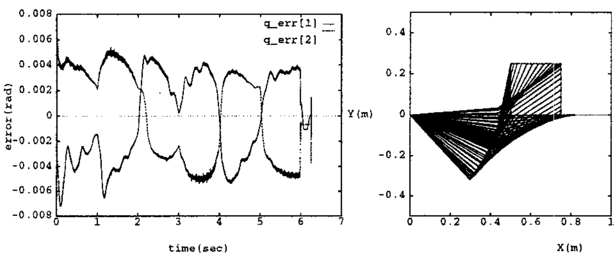
(b) Planar Two D.O.F Modular Manipulator

Fig. 3 Combined Modular Manipulator

2.4. Prototype System and Experiment

The developed modules can be assembled into some desired configurations. Fig. 3(a) shows one DOF vertical modular manipulator and Fig. 3(b) shows two DOF planar modular manipulator. This shows the main advantage of modular robot manipulator. We used planar two DOF modular manipulator for experiments. Joint position is measured by an incremental encoder which is directly attached to the motor shaft included in each joint module. It is controlled by simple independent joint control algorithm and the controller used for experiments is a pentium personal computer.

Fig. 4 shows arm motion for following square trajectory and joint motion error. Experimental result shows that it is operated with fairly good accuracy.



(a) Joint Position Error (b) Arm Motion

Fig. 4 Experiment for square trajectory following

3. Modular Robot Kinematics

For simplicity, we consider only two common types of revolute joint, rotate(R) and bending(B)(or pivoting) joints. They are distinguished by the orientation of the joint axis as mentioned before. Each module has to be connected sequentially and there is no offset. Even though redundant design of the manipulator is possible, we confine our attention to the minimal DOF design for a given task in this paper.

To determine necessary configuration of modular robot manipulator to perform given task trajectory, we need to introduce two coordinate system. One is attached to the base module, the other is attached to the end-effector. This coordinate system is depicted in Fig. 5. Generally, robot base coordinate ijk coincides with world coordinate XYZ. But if we attach some passive joint to the base module, then we can rotate the base coordinate system and it can be represented as an Euler angle. Firstly, we start with simple three-link manipulator(BBB) [1], and then according to the following kinematic conditions, we need to add rotary joint between the joints. Let's first determine the position of

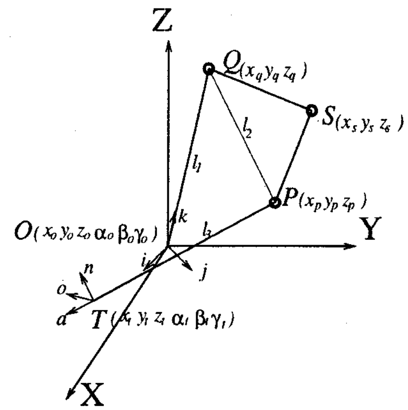


Fig. 5 Coordinate System of Modular Assembly

P and Q. This requires the knowledge of position and orientation of robot base and the end-effector which are given from task specifications. Then the position of Q and P is determined as follows.

$$Q(x_q, y_q, z_q) = l_1\vec{k} \quad (1)$$
$$P(x_p, y_p, z_p) = T(x_t, y_t, z_t) - l_3\vec{a} \quad (2)$$

According to the calculation result of Eqs. (3,4,5), addition of rotating joint 'R' is determined for the end-effector to track the given task trajectory.

$$\vec{j} \cdot \overrightarrow{PQ} \stackrel{?}{=} 0 \quad (3)$$
$$\vec{a} \cdot (\vec{k} \times \overrightarrow{QP}) \stackrel{?}{=} 0 \quad (4)$$
$$\vec{o} \cdot (\overrightarrow{PQ}) \stackrel{?}{=} 0 \quad (5)$$

For example, if dot product between $\vec{j}$ and $\overrightarrow{PQ}$ does not equal to zero, rotating joint 'R' is needed between O and Q point to correct the torsion angle. In the same way Eq. (5) means that if result of dot product doesn't equal to zero, rotating joint 'R' is also needed between T and P point to track the trajectory. Finally if the result of $\vec{a} \cdot (\vec{k} \times \overrightarrow{QP})$ is not zero, we can add another 'R' joint between Q and S point or between S and P point.

Through this procedure for sampled task trajectory, necessary joint configuration of manipulator can be determined from submanipulator of 'RBBRBR' or 'RBRBBR' [2]. From the selected configuration, we can obtain forward and inverse kinematics and dynamics. This data will be used to optimize the link length of the obtained manipulator structure in the next step of genetic algorithm.

[1] B means bending or pivoting joint, in Fig. 5 this basic structure is shown($OQSPT$),Q, S, P is the location of bending joint.

[2] RBBRBR is a manipulator which is offset free puma-type manipulator, RBRBBR is a reverse type of RBBRBR and this is a kind of minimal representation of the structure of nonredundant modular manipulator if we consider general task(three position and three orientation is required).

4. Genetic Algorithm

Genetic algorithm(GA) is a stochastic optimization algorithm that was originally motivated by the mechanisms of natural selection and evolutionary genetics[4]. In most conventional search technique, a single point is considered based on some decision rules, however GA works with a population of binary strings, searching many peaks in parallel. By employing genetic operators, they exchange information between the peaks, hence reducing the possibility of ending at local optimum and would be more likely to converge to the global optimum.

Table. 1 is the pseudo code of simple genetic algorithm [5] and explains the operation principle of GA. In this table, Pop. means population and Ind. means Individual.

Table 1 pseudo code of GA

procedure	Genetic Algorithm
begin	
initialize	Pop(k(generation) = 0)
evaluate	Ind. in Pop(k)
while	Terminal Condition **not satisfied, do**
begin	
	select Pop(k) from Pop(k-1)
	recombine Ind.in Pop(k)
	evaluate Ind.in Pop(k)
end	
end	

To initialize the procedure, the population is created by randomly selecting members from the allowable space. Then each of the individuals in the population are evaluated. Next, iterative loop which includes three basic operators begins. First operator is reproduction which is based on the law of survival of the fittest, in this routine some members of the population are selected for next generation based on their fitness. Selected members are recombined to form the population of the next generation by exchanging partial strings with each other. This procedure is called by crossover. This crossover allows combination of advantageous substructures into individuals more fit than their parents.

Mutation operator enhances the ability of GA to find a near-optimal solution. When used sparingly in combination with reproduction and crossover, mutation is an insurance policy against the loss of important genetic material at a particular position.

4.1. Determination of mutation probability and ranges of parameters

Mutation probability, crossover probability and ranges of parameters affect significantly on the efficiency of GA. So they are important and we propose a determination method of mutation probability and range of parameters as:

$$p = \left(\frac{i}{i_{max}}\right) p_{max} \tag{6}$$

$$(7)$$

$$\left.\begin{array}{rcl} x^j_{min} & = & \bar{x}^j - (\bar{x}^j - x^j_{min}) \exp\left(\eta \frac{i}{i_{max}}\right) \\ x^j_{max} & = & \bar{x}^j + (-\bar{x}^j + x^j_{max}) \exp\left(\eta \frac{i}{i_{max}}\right) \end{array}\right) \tag{8}$$

where η is the reducing rate, x^j is j^{th} parameter and $\bar{x}^j$is j^{th} fittest. p is the probability of mutation, i is a generation number and i_{max} is the maximum generation number. Eq. (6) shows how the mutation probability is determined. We see that the probability of mutation increases linearly as iteration proceeds. This equation enhances the ability of GAs to find a near-optimal solution. Ranges of parameters are determined by Eq. (8). The role of this equation is to reduce searching space efficiently to find an optimal solution because it raises the resolution of solution. Thus if we use the binary coding method, the link length l^j would be coded as binary strings of 0's and 1's with the bit length B^j for the parameters concerned with the resolution R^j. The bit length B^j and the corresponding resolution R^j is related by

$$R^j = \frac{x^j_{max} - x^j_{min}}{2^{B^j} - 1} \tag{9}$$

Eventually proposed determination method raises the resolution of solution as generation proceeds.

4.2. Fitness and Objective Function

The main role of an objective function in design is to choose one optimal design from the candidate which satisfies given constraints. For a dexterity measure to be used for design, it must be independent of the scale of a manipulator. In this paper, we use the relative manipulability as a dexterity measure for kinematic optimization problem. The relative manipulability is a dimensionless scale independent measure of manipulability and is defined by [2]

$$M_r = \frac{M}{f_M} \tag{10}$$

where M_r is the order independent manipulability and f_M is a function of dimension ([length]^2). The manipulability measure M is obtained as

$$M = \sqrt[m]{\boldsymbol{J}\boldsymbol{J}^T} \tag{11}$$

where $\boldsymbol{J}$ is the Jacobian matrix of instantaneous kinematics and m is the order of task space. For a 3 DOF planar manipulator with 2 dimensional task space, m = 2. We use l^2 for f_M, where l is a total length of a manipulator. In general, link length is defined as $(l_i = \sqrt{(a_i^2 + d_i^2)})$, where link length a_i,and offset d_i are D-H parameters. Since the task is given along trajectory, we use objective function root mean square of relative manipulability.

$$M_{rms} = \sqrt{\frac{\sum_k^{k_{max}} [M_{rk}{}^2 T_{sampling}]}{T_{completion}}} \tag{12}$$

where k_{max} is maximum work point, $T_{completion}$ is execution time of trajectory following, $T_{sampling}$ is $\frac{T_{completion}}{k_{max}}$.

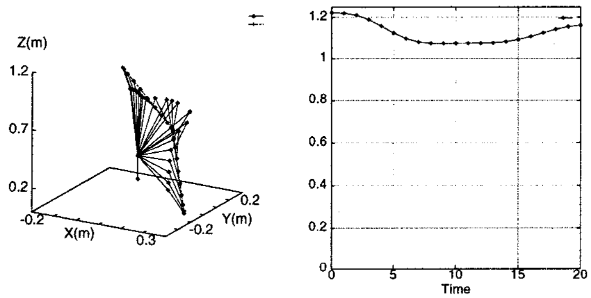
5. Design Examples

Now, by using the proposed two step algorithm, we try to find kinematic optimized modular manipulator configuration whose kinematics and objective function are described before. Since position and orientation of base is generally the same as world frame, we adjust the base coordinate ijk to the world frame XYZ.

Table 2 shows the task specifications requiring only position, and the number of task point is three. The desired trajectory is calculated by linearly interpolating between the given task points. Initial range of parameters is set from $0.1[m]$ to $0.6[m]$.

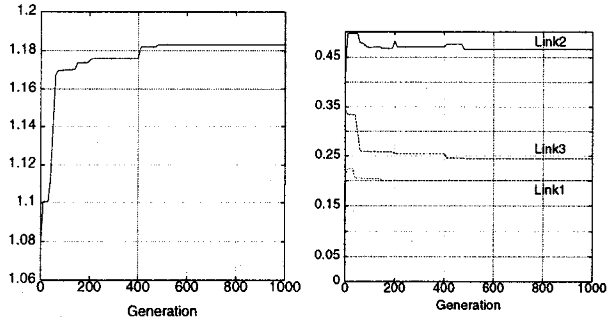
Table 2 Task Specification

TaskNum	p_x	p_y	p_z
1	0.5	-0.5	0.2
2	0.1	0.1	0.4
3	-0.3	0.4	0.6



(a) Arm Motion (b) Manipulability

Fig. 6 Optimized 3DOF Manipulator



(a) Maximum Fitness Link Length

Fig. 7 (b) Results of Optimized Manipulability

Table 3 Task Specification

TaskNum	p_x	p_y	p_z	θ_p	ϕ_p	ψ_p
1	0.2	0.4	-0.1	-300.0	200.0	300.0
2	0.1	0.3	0.1	-200.0	200.0	300.0
3	-0.1	0.2	0.2	-200.0	100.0	300.0
4	-0.2	-0.3	0.4	-200.0	100.0	200.0

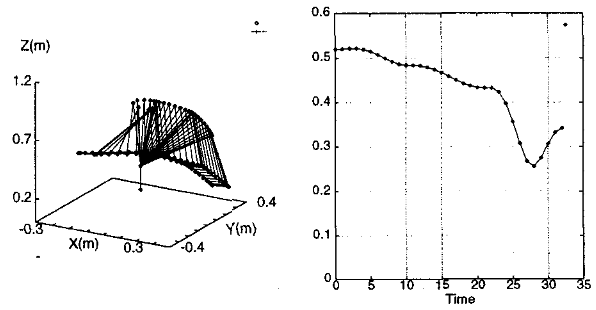
The results of optimized manipulator are shown in Figs. (6,7). The manipulator is chosen as a 'RBB' after the first step and its DOF is three [3]. Fig. (6)(a) shows the arm motion and (b) shows manipulability measure for an optimized 3 DOF manipulator. We can observe that resultant manipulability is maintained relatively high. Fig. (7) shows that the fitness(objective function return value) of parameter grows as generation increases and (b) shows the optimized link length l_1, l_2 and l_3 are approximately $0.2, 0.47$ and 0.245 respectively.

For planar two link robot, the manipulator takes its optimal configuration when $\theta_2 = \frac{\pi}{2}$, for any given values of l_1, l_2 and θ_1. But for puma type regional manipulator, if l_3 is smaller than l_2, its configuration is advantageous for near point from the first joint. On the contrary, if l_3 is larger than l_2, it is advantageous for far point from first joint [12]. In our case, optimized manipulator is the first case and the result is reasonable when we consider initial range of parameters.

Table 5. is the task specifications including orientation and the number of task point is four. The desired trajectory is also calculated by linearly interpolating between the given task points. Initial range of parameters is same preceding example.

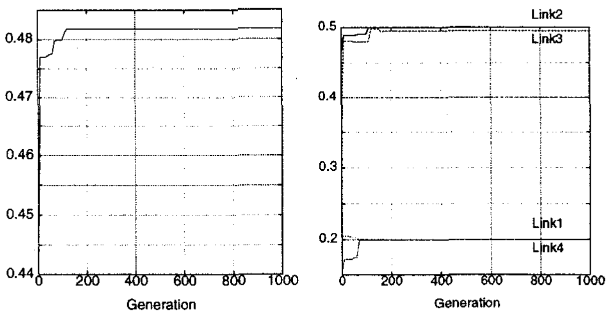
The results are shown in Figs. (8,9). The manipulator is chosen as a 'RBBRBR' [4] after the first step and its DOF 6. First link length l_1 and final l_4 is approaching to $0.2[m]$ and second and third link length is set

[3]RBB is like an offset free puma regional arm.
[4]RBRBBR is also possible



Arm Motion Manipulability

Fig. 8 Optimized 6 DOF Manipulator



Maximum Fitness Link Length

Fig. 9 Results of Optimized Manipulator

to $0.49[m]$ approximately. This result will be helpful in design and manufacturing each module for given task.

6. Concluding Remarks

This paper describes newly developed POSTECH modular manipulator and the method of task based design. The prototype includes two pivoting joint modules and one link module. New locking mechanism is proposed and it provides quick coupling and decoupling. The parallel connection is used to achieve small and lightweight design. It is tested by simple independent joint control algorithm and the experimental result shows that it is operated with fairly good accuracy.

We also proposed two step design algorithm : determination of robot configuration using kinematic relations and determination of link lengths using the proposed efficient genetic algorithm. The algorithm is used to search for an optimal manipulator from task specifications and the performance was evaluated for 3-DOF and 6-DOF manipulator.

Acknowledgment

This research was partially funded in part by KAERI(Korea Atomic Engineering Research Institute) under contract N95123.

REFERENCES

[1] D. Schimitz,P. K. Khosla and T. Kanade, "The CMU Reconfigurable Modular Manipulator System", Technical Report,Carnegie Mellon University,CMU-RI-TR-88-7,16pages,1988.

[2] J. O. Kim and P. K. Khosla, "Dexterity Measures for Design and Control of Manipulators", *IEEE/RSJ int. workshop on Intelligent Robots and Systems(IROS'91)*, Osaka,Japan,November,1991.

[3] J. O. Kim and P. K. Khosla, "A Multi-Population Genetic Algorithm and Its Application to Design of Manipulators", *IEEE/RSJ int. workshop on Intelligent Robots and Systems(IROS'92)*, July,1992.

[4] D. Goldberg,"Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley,1989.

[5] G. P. Roston, "A Genetic Methodology for Configuration Design", Carnegie Mellon University,Ph D. Dissertation, 1994.

[6] L. Kelmar and P. K. Khosla, "Automatic Generation of Kinematics for a Reconfigurable Modular Manipulator System", *IEEE Proc. of Int. Conf. on Robotics and Automation*,pp. 663–668,1988.

[7] T. Fukuda and S. Nakagawa, "Dynamic Reconfigurable Robotic System", *Proc. IEEE Conf. on Robotics and Automation*,Vol. 3,pp. 1581–1586,1988.

[8] K. H. Wurst, "The Conception and Construction of a Modular Robot System", *IFS,proc. Int. Symp. on Industrial Robotics(Belgium)*,pp. 37–44,1986.

[9] R. Cohen, M. G. Lipton,M. Q. Dai and B. Benhabib, "Conceptual Design of a Modular Robot", *Journal of Mechanical Design*,Vol. 114,pp. 117–125,1992.

[10] B. Benhabib,G. Zak and T. Kanade, "A Genealized Kinematic Modeling Method for Modular Robots", *Journal of Robotic Systems*,Vol. 6,No. 5,pp. 545–571,1989.

[11] D. Tesar and M. S. Butler, "A Generalized Modular Architecture for Robot Structures", *ASME Journal of Manufacturing Review*,DE-Vol.25,pp. 147–152,1989.

[12] T. Yoshikawa, "Foundation of Robotics : Analysis and Control", Reading, MA : MIT Press, 1990.

[13] T. Matsumaru, "Design and Control of the Modular Robot System : TOMMS", *IEEE Proc. of Int. Conf. on Robotics and Automation*, pp. 2125–2131, 1995.

[14] C. J. Paredis, P. K. Khosla, "Kinematic Design of Serial Link Manipulators From Task Specifications", *Int. J. Robotics Research*, 12, no. 3, pp 274–287,1993.

[15] I. M. Chen, "Theory and Applications of Modular Reconfigurable Robotic Systems", CALTECH,Ph D. Dissertation, 1994.