Proceedings of 2012 IEEE
International Conference on Mechatronics and Automation
August 5 - 8, Chengdu, China

Task-Based Configuration Synthesis for Modular Robot

Wenbin Gao[1,2] and Hongguang Wang[1]
Yong Jiang[1] and Xinan Pan[1,2]

1. *State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016*
2. *Graduate School of Chinese Academy of Sciences, Beijing, 100049*

***Abstract* - A modular robot system is introduced in this paper. The requirements for configuration synthesis are analyzed and classified as hard ones, soft ones and hard-soft ones. The evaluation function is constructed based on the weighted sum of the evaluation results of the specified design requirements. A one-lever Genetic Algorithm (GA) basing on an improved hybrid coding method is presented to carry out the configuration synthesis. In addition an example is given to demonstrate the effectiveness of this method.**

***Key words:* Modular Robot, Configuration Synthesis, Genetic Algorithm**

1. INTRODUCTION

The diversity and complexity of missions have becoming the trend of robotics. However, traditional robots sometimes couldn’t complete the issue of adaptation. For example, once a certain task isn’t taken into consideration in the robot initial design stage, it will have to design a new robot at this time. Modular Robotic Systems (MRSs) are proposed to solve this problem, which comprised of a serious of standard modules such as link modules, joint modules and gripper modules with different dimensions and certain assembly styles [1-3].

The determination of a feasible topology meeting the functional requirements is a multi-objective optimization problem by saying: determine the module types, ports and assembly orientations from a given set of modules to construct the most suitable task-oriented robot topology. Progressive methods try to realize the optimized solutions through matching up with different requirements iteratively. Unfortunately, the strongly coupled requirements imply numerous backtrackings, which would slow down the search process and even hinder from getting optimal solution. Global methods (such as Genetic Algorithm (GA)) which consist in trying to meet all requirements simultaneously are widely adopted in modular robot configuration syntheses since the robust property to local minimal and search spaces with highly nonlinear[4-6].

Evaluation function which is the foundation of configuration optimization can be constructed in two ways. The first way is to classify the requirements as optimization goal and constraints. The optimization goal is taken to evaluate the given topologies and the constraints are regarded as filters [7-9]. The second way is to evaluate the requirements successively and give weighted values to the evaluation results according to the importance in the applications. Then the objective function is constructed using weighted sum of the evaluation results [4,10-12]. Both methods have their own limitations. Because some of the requirements should be taken as optimal objectives and filters at the same time.

The paper is organized as follows. In section 2: a Modular Reconfigurable Robot Experiment System (MRRES) is introduced. An improved coding method for configuration synthesis is presented. In section 3: the design requirements are analyzed and the configuration synthesis process is given. In section 4: an example is presented. Finally, the conclusion is given in the last section.

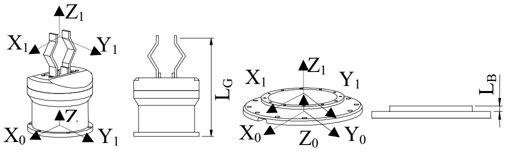
2. MRRES

The MRRES (Fig.1) is developed by the State Key Laboratory of Robotics in Shenyang Institute of Automation, China. It takes a distributed control system with an Industrial Personal Computer as the main controller and DSPs in the joint modules. The data communication is based on CAN bus.

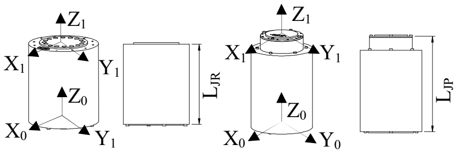
Fig. 1 MRRES

2.1 Module library of MRRES

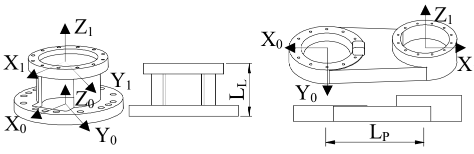
MRRES’s module library contains revolute joint modules, prismatic joint modules, right-angled link modules, parallel link modules, linear link modules, base modules and gripper, shown in Fig.2.



(a) Gripper and Base modules

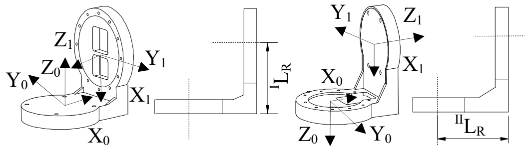


(b) Revolute joint and Prismatic joint



(c) Linear link and Parallel link

978-1-4673-1278-3/12/$31.00 ©2012 IEEE



(d) I and II styles of Right-angled links

Fig.2 Modules of MRRES

2.2 Configuration coding

The information of module topology and joint displacements is the bond between the design requirements and design variables. Individual coding for topology and joint displacements is a base for GA to implement the configuration synthesis. In this paper, the joint types and their assemble sequences aren't involved in the topology coding due to the coupling between the joint types and the joint displacements. As shown in Fig.3, if the corresponding joint types in the same position of two individual topologies are different, the joints variables' crossover operation can't be reasonably implemented. In the actual design applications, the number of DOFs for a manipulator can be specified directly according to the tasks. The number of reasonable proportion and location of the prismatic joints in a manipulator with given DOFs is limited. So the joint types and locations could be assigned based on the experience prior to synthesis process.

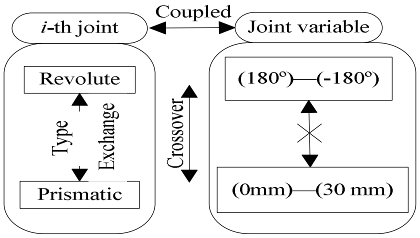


Fig.3 Coupling between the joints' types and variables

2.2.1 Topology coding

The topology information which is discrete variables should be coded in binary form. The coding and decoding methods for the module lengths (Fig.2) are shown in Table1.

Table.1 Coding for the model lengths

Type	Binary coding	Length(mm)
L_B	3-bit	$0+5n$
L_L	3-bit	$40+5n$
L_P	3-bit	$180+5n$
$^{I}L_R$	3-bit	$89.5+5n$
$^{II}L_R$	3-bit	$89.5+5n$

1). n is the decimal representation of the binary coding;

2). L_B, L_L, L_P, $^{I}L_R$, $^{II}L_R$ represent both the module types and their lengths, when the confusions aren't caused.

The assembly orientation of two assembled modules is defined as: the input port of the latter module turning about the z-axis of output port of the former module in clockwise direction. The coding method for the assembly orientation and the link type is listed in Table2.

A binary coding example for the base assembly is indicated in Fig.4. A binary coding example for a sub-assembly including joint and link modules is graphed in Fig.5

Table.2 Coding for assembly orientations and link types

Orientation	0°		90°		180°		270°	
Coding	0	0	0	1	1	0	1	1
Link type	L_C		L_P		$^{I}L_R$		$^{II}L_R$	
Coding	0	0	0	1	1	0	1	1

2.2.2 Configuration coding

It has been shown that in term of continuous searching space, real coding algorithm is more suitable than binary coding algorithms. So in the posture calculation process, the joint displacements which are real variables should keep the 'real-coded' form [13].

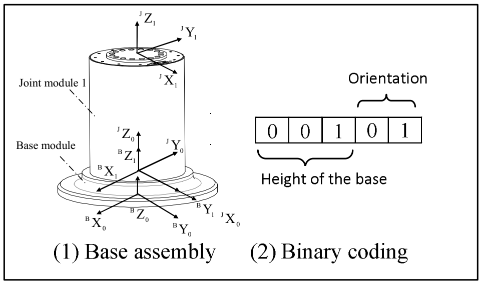


Fig.4 Chromosome segment of the base assembly

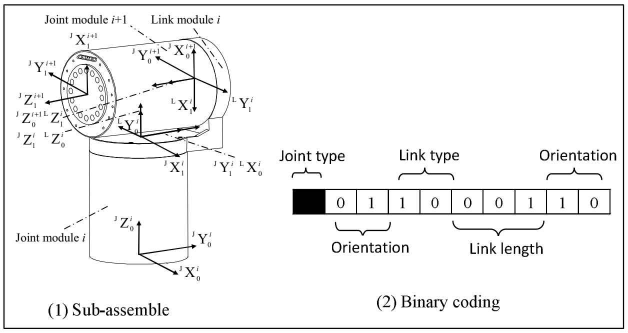


Fig.5 Chromosome segment of a sub-assemble

The GA based configuration synthesis methods can be implemented in two ways. The first one is in the tow lever form which takes the upper lever to calculate the topology and the lower lever to calculate the posture. The shortcoming is that it is usually a time-consuming method. An alternative method is to combine the binary coding and real coding as a mix-coded method. In this way, the topology and posture information can be represented by one chromosome individual. Then the whole configuration synthesis process can be carried out by a one-lever GA [4,8,11].

Based on the analysis and definition above mentioned, the binary-real coding for an n-DOFs configuration is as follows.

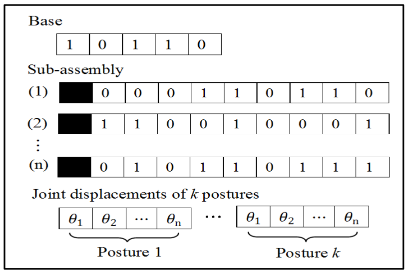


Fig.6 Binary-real coding for a configuration

2.2.3 Forward kinemics

As shown in Fig.2, frames are set on the input port (subscript 0) and output port (subscript 1) of each module respectively. According to the coding strategy given in section 2.2.1, the module types, parameters and assembly orientations can be got from the chromosome of an individual. The position and orientation of the frame on the *i*th joint's input port expressed in the base frame is

$$\boldsymbol{T}_i = \begin{cases} \boldsymbol{T}_B \, {}_{\mathrm{B}}\boldsymbol{T}_i^{\mathrm{J}} & i = 1 \\ \boldsymbol{T}_{i\text{-}1} \boldsymbol{T}_{i\text{-}1}^{\mathrm{J}} \, {}_{\mathrm{J}}\boldsymbol{T}_{i\text{-}1}^{\mathrm{L}} \boldsymbol{T}_{i\text{-}1}^{\mathrm{L}} \, {}_{\mathrm{L}}\boldsymbol{T}_i^{\mathrm{J}} & i = 2 \cdots n \end{cases} = \begin{bmatrix} \boldsymbol{R}_i & \boldsymbol{p}_i \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_i & \boldsymbol{o}_i & \boldsymbol{\omega}_i & \boldsymbol{p}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Where, when all joints are at zero position, $\boldsymbol{T}_{\mathrm{B}}$, $\boldsymbol{T}_i^{\mathrm{J}}$, $\boldsymbol{T}_i^{\mathrm{L}}$ are the homogeneous transformation matrices from the output frames to the input frames for the base module, *i*th joint module, *i*th link module respectively. ${}_{\mathrm{B}}\boldsymbol{T}_1^{\mathrm{J}}$, ${}_{\mathrm{J}}\boldsymbol{T}_i^{\mathrm{L}}$, ${}_{\mathrm{L}}\boldsymbol{T}_i^{\mathrm{J}}$ describe the assembly orientations of modules relative to their former modules. For example, ${}_{\mathrm{B}}\boldsymbol{T}_1^{\mathrm{J}}$ describes the orientation of the first joint relative to the base module.

The position and orientation of the output frame of the gripper expressed in the base frame is

$$\boldsymbol{T}_{n+1} = \boldsymbol{T}_n \boldsymbol{T}_n^{\mathrm{J}} \, {}_{\mathrm{J}}\boldsymbol{T}_n^{\mathrm{L}} \boldsymbol{T}_n^{\mathrm{L}} \, {}_{\mathrm{L}}\boldsymbol{T}^{\mathrm{G}} T^{G} \tag{2}$$

Where, ${}_{\mathrm{L}}\boldsymbol{T}^{\mathrm{G}}$ describes the orientation of the gripper relative to the former link. $\boldsymbol{T}^{\mathrm{G}}$ is the homogeneous transformation matrix of gripper from the output frame to the input frame.

As for revolute joints, the screw is

$$\boldsymbol{\xi}_i = \begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\omega}_i \times \boldsymbol{p}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \tag{3}$$

For prismatic joints, the screw is

$$\boldsymbol{\xi}_i = \begin{bmatrix} \boldsymbol{\omega}_i \\ 0 \end{bmatrix} \tag{4}$$

And then the forward kinematics of a given topology of modular robot is [14]

$$\boldsymbol{g}_{\mathrm{BG}}(\boldsymbol{\theta}) = e^{\hat{\xi}_1 \theta_1} e^{\hat{\xi}_2 \theta_2} \cdots e^{\hat{\xi}_n \theta_n} T_{n+1} \tag{5}$$

3. EVALUATION FUNCTION

The design requirements can be classified into two levers: module lever and manipulator lever, as shown in Fig.7. A feasible topology of a given MRS must meet a serial of requirements. The requirements can be divided into three types: hard ones, soft ones and hard-soft ones. The hard requirements are that must be satisfied. The soft ones haven't rigid targets. The synthesis progress is implemented to find out an optimal solution. The hard-soft requirements are conducted to look for an optimal solution and fulfill the hard requirements at the same time. Whether a requirement is a hard one, soft one or hard-soft ones is changeable according to the user's requirements. Taking the power consumption for example, if we give a determined power limit it will become a hard requirement. If we only require the robot's power consumption lower, it will be a soft requirement. If the power consumption not only needs to satisfy the limit but also be as lower as possible, then it will be a hard-soft requirement.

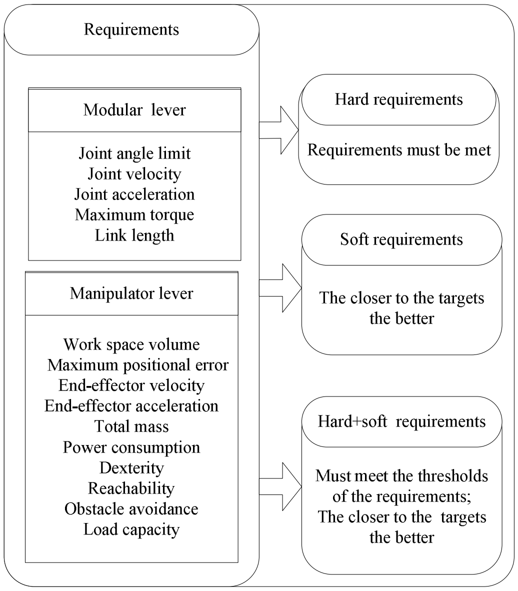


Fig.7 Design requirements

3.1 Requirements evaluation

According to the application needs, the requirements exerted on the configuration synthesis for MRRES are as follows.

3.1.1 Requirements on module lever

(1) Joint angle requirement

Joints' angle limit always belongs to the hard requirements. The limit of *i*th joint can be expressed as

$$\theta_i^{\min} \le \theta_i \le \theta^{\max}_i \tag{6}$$

The joint variables can be got by solving the inverse kinematics based on the GA method. The interference with the joints' limits which may be occurred in the crossover or mutation process will be deal with by a filter. If the joint displacement is between the limit and a given threshold, then a soft constrain is exerted. Otherwise, if the value is less than the threshold, the soft constrain will be quitted, as follows

$$E_{\theta i} = \begin{cases} exp(K_i) - 1, & 0.5 \le K_i < 1 \\ 0, & K_i < 0.5 \end{cases} \tag{7}$$

Where, $K_i = 1 - \dfrac{\left|\theta - \left(\theta_i^{max} + \theta_i^{min}\right)/2\right|}{(\theta^{max}_i - \theta_i^{min})/2}$ is a normalization process for both the revolute and prismatic joints. We can find that $E_{\theta i}$ would run an exponential growth in case of $0.5 \le K_i < 1$, which will keep the joint variable away from limit quickly.

The sub-evaluation function about joint displacements is

$$f_\theta = w_\theta \sum_{i=1}^{n} E_{\theta i} \tag{8}$$

Where, $w_\theta = 1$ is the weighted number.

3.1.2 Requirements on manipulator lever

(1) Zero position requirement

A manipulator is expected to be upright to lessen the effect of gravity when it is in a non-working state. The constrain can be simply implemented as

$$f_z = w_{Ez} E_z = w_{Ez}\sqrt{\left\| p_x^2 + p_x^2 \right\|} \tag{9}$$

Where, p_x, p_y are the x-component and y- component of the zero position of the gripper relative to the base frame respectively. $w_{Ez} = 0.1$ is the weighted number.

(2) Link type requirement

In some cases, a given topology of MRRES may lose several DOFs. This is a situation which should be avoided unless redundancy is a design need. In order to avoid this, a topology with n-DOFs should meet the following 4 rules:

i The last link connecting the last joint and gripper is allowed to be a linear link regardless of the type of the last joint.

ii As shown in Fig.8, two revolute joints couldn't be connected by a linear link;

$$E_R = \sum_{i=1}^{n-1} J_i L_i J_{i+1} \tag{10}$$

Where, $J_i = \begin{cases} 1 & \text{Revolute joint} \\ 0 & \text{Prismatic joint} \end{cases}, L_i = \begin{cases} 1 & \text{Linear link} \\ 0 & \text{Other link} \end{cases}$.

iii For each joint, linear link is allowed only to be assembled to one of the two ports: input port or output port;

$$E_J = \sum_{i=1}^{n-2} L_i L_{i+1} \tag{11}$$

iv Two prismatic joint shouldn't be connected by a linear link.

$$E_p = \sum_{i=1}^{n-1} (1-J_i) L_i (1-J_{i+1}) \tag{12}$$

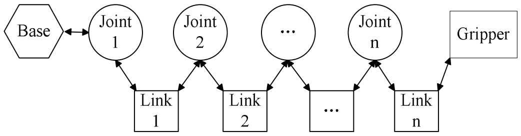


Fig. 8 Sketch map for topology constrains

The sub-evaluation function about topology requirements can be described as

$$f_{topo} = w_{topo}\left(E_R + E_J + E_P\right) \tag{13}$$

Where, $w_{topo} = 5$ is the weighted number. If $\left(E_R + E_J + E_P\right) \geq 6$, then the topology will be deal with by a filter.

(3) Reachability requirement

The linear distance between the target position and the actual position of the gripper is

$$E_L = \left\| p_t - p_a \right\| \tag{14}$$

$$w_L = \frac{1}{1.7 p_0} = \frac{1}{1.7\sqrt{\left\| p_x^2 + p_y^2 + p_z^2 \right\|}} \tag{15}$$

Where, p_t and p_a are the target and actual positions respectively. w_L is the weighted number. p_0 is the distance between the gripper and the base frame.

The orientation distance between target orientation and actual orientation of the gripper is[14].

$$E_O = \left\| O_t - O_a \right\| \tag{16}$$

$$w_O = \frac{1}{2\pi} \tag{17}$$

Where, O_d and O_e are the Euler angles of the target orientation and the actual orientation respectively. w_O represents the weighted number.

The sub-evaluation function about reachability requirements can be described as

$$f_{reach} = w_L E_L + w_O E_O \qquad 0 < K_R < 1 \tag{18}$$

Where, $K_R = \frac{d_{max}}{l_0}$, $l_0 = \sum_{i=1}^{n} l_i$ is the approximate lengths of the robots when it's in zero position; l_i represents the length of the modules. d_{max} represents the distance from a work point to the robot base. If $K_R \geq 1$, a filter is imposed.

3.2 Evaluation function for MRRES

According to the requirements described above, the manipulator evaluation function can be given as

$$f(Chrom) = f_\theta + f_z + f_{topo} + f_{reach} \tag{19}$$

The configuration synthesis process is shown in Fig.9. If a candidate configuration interferes a hard constrain, then it will be deal with by the filters. Otherwise, it will be evaluated by the evaluation function.

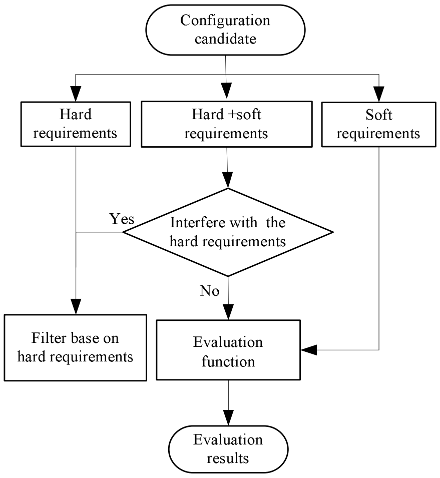


Fig.9 Configuration evaluation procedure

4. EXAMPLE

4.1 Task description

A robot's task usually can be described as a collection of working points that the end- effector must be realized. When the task is to follow a trajectory, it can be approximated by a set of points along the path. In this example, the task is described as 3 working points, shown in Table.3.

Table.3 Target postures of the end-effector

	Euler angle(º)			Position(mm)		
Target 1	-23.08	64.18	-4.31	720.94	-450.18	808.86
Target 2	-54.86	87.53	11.95	652.27	-761.70	453.78
Target 3	-44.03	65.87	31.06	706.54	56.70	666.82

4.2 Joints assignation

According to the Section 2.2, the joint type, number and location should be assigned before the configuration optimization, as follows

Table.4 Joint type and arrangement

Number	1	2	3	4	5	6
Type	0	0	1	0	0	0

1). 0 means revolute joint;
2). 1 means prismatic joint.

4.3 Parameters of GA

The number of individuals of the populations is 800. The number of generations is 600. The selection, crossover, mutation operations are described as follows

(1) The selection operation is taken out with stochastic universal sampling according to the fitness of the population. The number of the result population is 80% of the initial population.
(2) The crossover operation randomly selects two parents from the parent pool. The binary parts are imposed crossover operation with probability 0.7 and the real parts are experienced an intermediate recombination separately. Then the two children are produced by recombining the two parts.
(3) The mutation operation is exerted on both the binary parts and real parts, the mutation rates are 0.012 and 0.17 respectively.

4.4 Configuration synthesis

The configuration synthesis procedure is shown in Fig.10. The objective values calculated based on the evaluation function in section 3.2 are ranked from large to small. The individual fitness is assigned linearly in [0 2] according to the order of the objective values. The individuals which conflict with the hard requirements will be deal with by the filters. In this implementation, the filters are not carried out to remove an individual directly, but give it a fitness small enough directly and stop the further evaluation. In the actual synthesis process, it usually have no change to propagate its own offspring, in other words it is filtered.

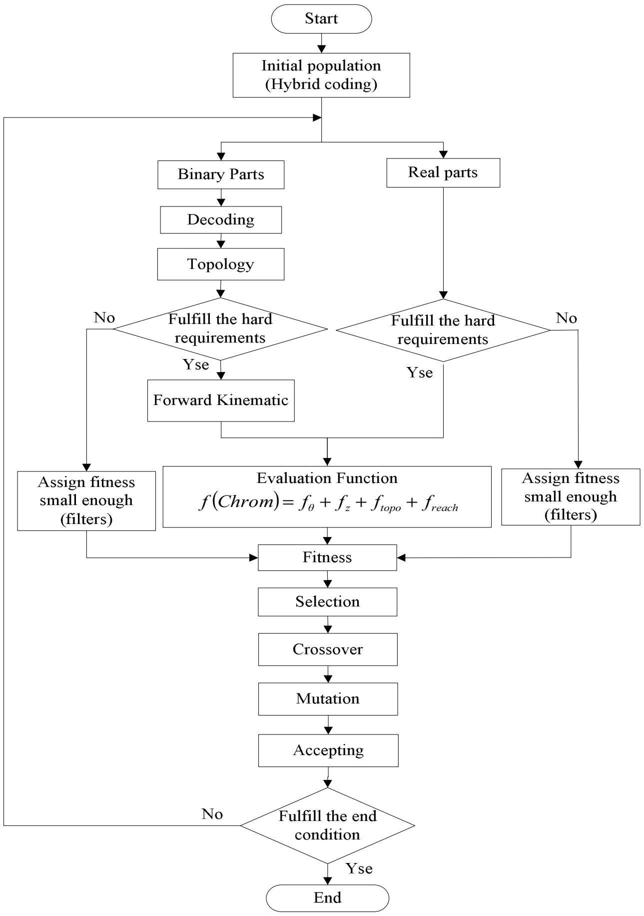


Fig.10 Configuration synthesis procedure

4.5 Result

The configuration synthesis results are given in this sub-section. It is showed in Fig.11 that the minimum evaluation results respect to the generation numbers. The optimized topology with 6-DOF is shown in Fig.12(a). The corresponding binary coding and the decoding result are indicated in Table.5.

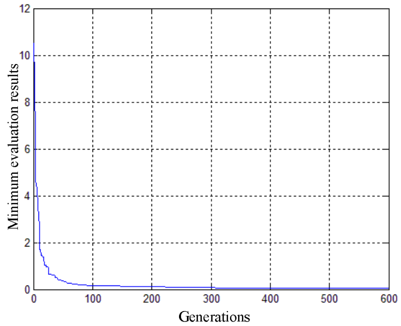


Fig.11 Minimum evaluation results respect to the generation numbers

The three result poses of the end-effector are shown in Table.6. The corresponding joint variables are listed in Table.7. Fig.12(b-d). shows the result postures of the manipulator.

We can see from Table.3 and Table.6 that there are some errors between the target and result poses of the end-effector.

But the errors are usually tolerable for configuration synthesis. High-precision inverse kinematics calculation for the result topology can be implemented in the path planning process.

Table.5 Result of the topology synthesis

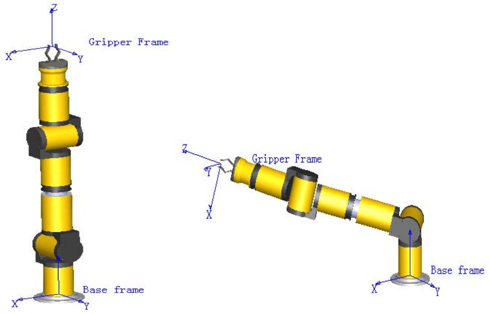
		Orientation (º)		Link type		Length (mm)			Orientation (º)	
Base	Coding					**0**	0	1	1	0
	Decoding			L_B		5			180	
Link （1）	Coding	0	1	1	0	0	0	0	1	0
	Decoding	90		$^{I}L_R$		71.5			180	
Link （2）	Coding	0	0	1	1	0	0	0	1	0
	Decoding	0		$^{II}L_R$		71.5			180	
Link （3）	Coding	1	1	0	0	0	0	0	1	0
	Decoding	270		L_C		40			180	
Link （4）	Coding	0	0	1	0	0	0	0	0	1
	Decoding	0		$^{I}L_R$		71.5			90	
Link （5）	Coding	0	1	1	1	0	0	0	0	0
	Decoding	90		$^{II}L_R$		71.5			0	
Link （6）	Coding	0	1	0	0	0	0	0	0	1
	Decoding	90		L_C		40			90	

Table.6 Result poses of the end-effector

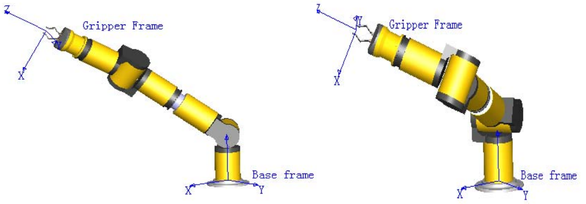
	Euler angle(º)			Position(mm)		
Result1	-24.57	60.09	-4.81	727.84	-451.41	808.34
Result2	-58.78	79.42	12.16	652.65	-761.90	453.61
Result3	-43.53	67.11	31.13	706.53	56.68	666.94

Table.7 Result joint variables

Joint type	0(º)	0(º)	10(mm)	0(º)	0(º)	0(º)
Displacement	-37.85	53.70	8.13	-26.13	12.81	28.60
Displacement	-42.11	76.83	9.31	7.24	-16.52	1.47
Displacement	40.38	64.47	13.13	-18.63	-75.18	9.28



(a) 6-DOF topology (b) Result posture 1



(c). Result posture 2 (d). Result posture 3

Fig.12 Result topology and postures of the manipulator

5. SUMMARY

A modular robot system named MRRES is presented. The fact that the requirements of both the modular lever and manipulator lever could be classified as hard ones, soft ones and hard-soft ones according to the applications is pointed. Expert knowledge about topology design is discussed and presented as a design requirement. An improved hybrid binary-real coding method is presented: joints' number, types and sequence are assigned to simplify the crossover and mutation operations; binary coding is taken to represent the topology and the real coding to represent the joint variables. A one-lever GA is taken to conduct the configuration synthesis. The effectiveness of this method had been proved by an example.

ACKNOWLEDGEMENTS

The research of this paper is supported by the National Natural Science Foundation of China (Grant No. 60905048) and the National Hi-tech Research and Development Program of China (863 Program, Grant No. 2007AA041703).

REFERENCES

[1] WenHong Zhu, Tom Lamarche, Patrick Barnard, "Modular Robot Manipulators with Preloadable Modules," IEEE International Conference on Mechatronics and Automation, Harbin, China, 2007, 7-12,

[2] Matsumaru T, "Design and Control of the Modular Robot System: TOMMS," IEEE International Conference on Robotics and Automation, Nagoya, Japan, vol.2, 1995, 2125- 2131

[3] Ming Chen, Yan Gao, "Closed-Form Inverse Kinematics Solver for Reconfigurable Robots," IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001, 21-26

[4] Chocon, P. Bidaud, "Genetic Design of 3D Modular Manipulators," IEEE International Conference on Robotics and Automation, New Mexico, USA, 1997, 223-228

[5] Christiaan J. J. Paredis, Pradeep K. Khosla, "Synthesis Methodology for Task Based Reconfiguration of Modular Manipulator Systems," 6th International Symposium on Robotics Research. Hidden Valley, PA, 1993

[6] Leger Chris, Bares John, "Automated Task-Based Synthesis and Optimization of Field Robots," Robotics Institute, http://repository. cmu.edu /robotics /91, 1999, 91

[7] I-M Chen, Joel Burdick, "Determining Task Optimal Modular Robot Assembly Configurations," IEEE International Conference on Robotics and Automation, 1995, 132-137

[8] Guilin Yang, I-Ming Chen, "Task-Based Optimization of Modular Robot Configurations: Minimized Degree-Of-Freedom Approach," Mechanism and Machine Theory , vol.35, 2000, 517-540

[9] Jeongheon Han, W.K.Chung, Y.Youm. et al, "Task Based Design of Modular Robot Manipulator using Efficient Genetic Algorithm," IEEE International Conference on Robotics and Automation, New Mexico, USA, 1997, 507 -512

[10] Chocon, P. Bidaud, "Evolutionary Algorithms in Kinematic Design of Robotic Systems," IEEE/RSJ International Conference Intelligent Robots and Systems, Grenoble, France, 1997, 1-7

[11] Chocron, "Evolutionary Design of Modular Robotic Arms," Robotica, vol.26, 2008, 323–330

[12] Wei Yanhui, Zhao Jie, Cai Hegao, "Task-Based Method for Determining Topology of Reconfigurable Modular Robot," Chinese Journal of Mechanical Engineering, vol.5(42), 2006, 93-97

[13] Saleh Tabandeh, Christopher Clark, William Melek, "A Genetic Algorithm Approach to Solve for Multiple Solutions of Inverse Kinematics using Adaptive Niching and Clustering," IEEE Congress on Evolutionary Computation, BC, Canada, 2006, 1815-1822

[14] Richard M. Murry, Zexiang Li, S. Shankar Sastry, "A Mathematic Introduction to Robotic Manipulation," CRC Press,1994