CARV 2007

Task-based Configuration Optimization of Modular and Reconfigurable Robots using a Multi-solution Inverse Kinematics Solver

S. Tabandeh[1], C. Clark[2], W. Melek[3]

Abstract: Modular and Reconfigurable Robots (MRR) are a breed of industrial robots designed for today's flexible and versatile production facilities. MRRs can be assembled in various ways to achieve numerous distinct Kinematic configurations (KC). In MRRs, the problem of finding the most suitable KC for a specific task from a predefined set of modules is called Task-based Configuration Optimization (TBCO). To assess a KC in TBCO, a solution to the Inverse Kinematics (IK) problem is required.

In this paper, a TBCO algorithm based on Genetic Algorithms (GA) is proposed that utilizes a multi-solution IK solver. By solving for multiple solutions of the IK, The possibility of preferring a sub-optimal KC over an optimal configuration is decreased. Moreover, in the proposed TBCO algorithm a priority based GA Selection operator is used to ensure reaching the right solution in case of more than one optimization criteria.

Keywords: Modular and Reconfigurable Robots, Task-Based Configuration Optimization, Genetic Algorithms, Inverse Kinematics

1 Introduction

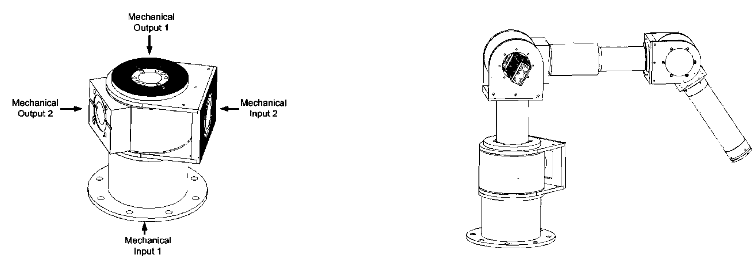
A main building block of many automated manufacturing plants is the industrial robot manipulator. The majority of the existing industrial robots are based on a

[1] University of Waterloo, Mechanical Engineering Department

[2] California Polytechnic State University, Department of Computer Science

[3] University of Waterloo, Mechanical Engineering Department

fixed joint and link configuration and are designed to perform their duties for a general set of tasks. Although these robots can perform well in a set of particular workspaces, they have limited adaptability towards changes in either environment or the prescribed task. MRRs , on the other hand, are assembled from a variety of modular components and can be physically configured to meet the requirements of the work space and the task at hand. The set of modules may consist of joints, links, and end-effectors. Joint modules are the actuators that provide the degree of freedom of each robot. Link modules of varying length connect the joints to each other. The end-effector modules consist of the tools required to interact with the robot environment. Figure 1 shows the schematic diagram of a joint module and an MRR that is assembled into a three degree-of-freedom (DOF) robot. As can be seen from the figure, the joint module has two mechanical input and two mechanical output ports. Input ports should be connected to the links closer to the base of the robot while the output ports should be connected to the links closer to the end-effector. The joint will produce different types of motion depending on which input and ouput ports the links are connected to.



a b

Figure 1) a. The schematic of a Joint module with two mechanical inputs and two mechanical outputs b. An MRR assembled into a 3 DOF KC

Different KCs can be achieved by using different joint, link, and end-effector modules and by changing their relative orientation and, in the case of the illustrated example, the joint ports used. The number of distinct KCs attainable by a set of modules can vary with respect to the size of the module set from several tens to several thousands.

Finding the most suitable configuration for a specific task from a predefined set of modules is a highly nonlinear optimization problem in a discrete search space called TBCO.

In the literature, different approaches to solve the have been presented.

In (Chen 1994) Chen introduced a new representation called Assembly Incident Matrix (AIM) for Modular Robots. AIM is then translated to a string and is used in a Genetic Algorithm, with reachability and manipulability as the objective functions. In (Chen, *et al.*, 1998), the same concept is expanded to make a modified AIM that includes the port vectors. The mutation and crossover operators of the Genetic Algorithm are also modified to be directly applicable to the AIMs. In (Leger 1999) an object orientated structure to represent robots is used with a GA. In some cases, the algorithm relies on a human expert to redirect the search. In (Kim 1992) Task based configuration optimization problem is tackled with the addition of numerous optimization criteria. A two level Genetic Algorithm approach is used in (Chocron, *at al.*, 1997). The upper level GA searches for the most suitable configuration and the lower GA solves for a single solution to the Inverse Kinematics.

Most of the existing TBCO algorithms use a GA to optimize the KC. At each iteration of these GAs, a solution to the Inverse Kinematics problem is required. The approaches referenced mostly rely on numerical methods to solve the IK problem. Unfortunately, numerical IK methods only converge on the solution that is closest to their starting point. Therefore, only one solution of the Inverse Kinematics will be considered when two KCs are being compared.

In this paper, a TBCO algorithm is proposed that uses a Genetic Algorithm as the core optimization engine (as in previous works). However, in contrast to the existing methods the proposed algorithm utilizes multiple solutions of the IK problem. To facilitate that, an IK solver capable of calculating multiple solutions of the IK for a wide range of robotic manipulators has been developed.

In summary the contributions of this paper can be highlighted as:

- A novel MRR representation called KMR is proposed.
- A multi-solution Inverse Kinematics solver is used for assessing the KCs.
- A novel priority base Tournament Selection operator specifically modified for TBCO is proposed. This operator enables the algorithm to search for a KC that can perform a certain task better than the others according to an efficiency criterion.
- TBCO specific Crossover and Mutation operators have been proposed. By using the proposed operators, The GA is applied directly on the KMR without the necessity of coding to a string.

This paper is organized as follows. In Section 2 the proposed TBCO algorithm is explained in detail. The subsections include problem formulation, Module inventory and kinematic representation and the Inverse Kinematics. In Section 3 the results of applying the proposed TBCO algorithm in finding a 3DOF kinematic configuration for a task is presented. Section 4 discusses the conclusion and the future work of the project.

2 Task Based Configuration Optimization

The goal of TBCO is finding a KC, or an assembly sequence of the modules, capable of performing a certain task more efficiently than the other configurations.

A task in the Cartesian space is defined by a set of task points. Each task point consists of a position and an orientation in space. The vector of the position of task point *t* with respect to the origin is represented by P_{OT}^{t} and is shown in Eq.(1.1). The orientation of task point *t* is represented by O_{OT}^{t} and can be defined as in Eq.(1.2). In this equation, $(\alpha_t, \beta_t, \gamma_t)$ are the Euler angles of the task point *t*.

$$P_{OT}^{t} = \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} \tag{1.1}$$

$$O_{OT}^{t} = \begin{pmatrix} \alpha_t \\ \beta_t \\ \gamma_t \end{pmatrix} \tag{1.2}$$

The measure that represents the ability of a KC to accomplish a task is called reachability. Reachability shows how capable a KC is in reaching the position and orientation of the task points. Position reachability for task point *t* is represented by $f_{rch,t}^{p}$ and can be defined as in Eq.(1.3).

$$f_{rch,t}^{p} = \left\| P_{OT}^{t} - P_{OE} \right\| \tag{1.3}$$

Where P_{OE} is the position of the end-effector of KC when it is as close as possible to P_{OT}^{t}. In more specific terms, f_{rch}^{p} is the distance of the position of the task point to that of the KC when the KC is as close as possible to the task point. Eq.(1.4) shows the orientation reachability definition.

$$f_{rch}^{o} = \min(\left\|(\alpha_t, \beta_t, \gamma_t) - (\alpha_e, \beta_e, \gamma_e)\right\|, \left\|(\alpha_t, \beta_t, \gamma_t) - (\alpha_e + \pi, -\beta_e, \gamma_e - \pi)\right\|) \tag{1.4}$$

Where $(\alpha_e, \beta_e, \gamma_e)$ are the orientation of the end-effector when it is as close as possible to that of the task point. The total reachability can be defined as the weighted sum of f_{rch}^{p} and f_{rch}^{o} as in Eq.(1.5) Where w_{rch}^{p} and w_{rch}^{o} are the weigthing factors.

$$f_{rch} = w_{rch}^{p} . f_{rch}^{p} + w_{rch}^{o} . f_{rch}^{o} \tag{1.5}$$

A position and orientation reachability of zero for a certain task indicates that the KC is capable of performing the task. In practice, usually more than one KC exist that can accomplish a certain task. Hence, the KC that can perform the task more efficiently than the others according to an optimization criterion (or a set of criteria) should be sought.

The TBCO can be formulated into an Optimization Problem as in Eq.(1.6).

$$\begin{cases} \min\limits_{R \in \mathrm{K}} \quad f_{op} \\ \text{Subject to}\, f_{rch} \le \varepsilon \end{cases} \tag{1.6}$$

As can be seen from the equation, the goal is finding a Kinematic Configuration R capable of achieveing the task within a reachability less than a tolerance value ε while minimizing an optimization criteria f_{op}. In this reaseach, as the optimization criterion f_{op}, Relative Manipulability is used. K represents the space of all the possible kinematic configurations attainable by a set of modules.

Figure 2 shows the architecture of the proposed TBCO algorithm. The inputs to the algorithm are module inventory, optimization criteria, and the task. These inputs must be coded before entering the algorithm. The way these parameters are coded greatly depends on the optimization algorithm that is used, and has a dramatic effect on the performance of the algorithm.

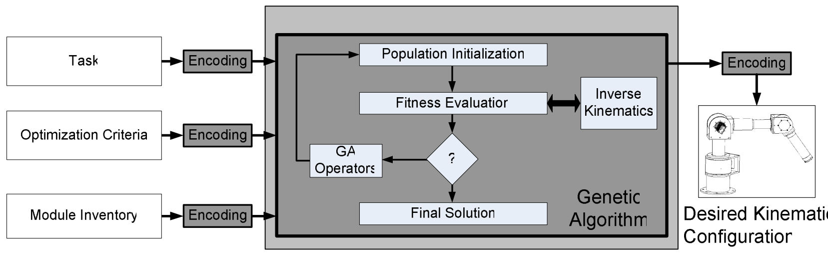


Figure 2) Architecture of the proposed TBCO algorithm

The first set of inputs, module inventory, is the group of available modules. This set determines the size and complexity of the Kinematic Configuration space K . The optimization parameters are another group of inputs to the algorithm. These parameters are application dependent and the set of parameters relevant to the tasks are highlighted and selected as the inputs of this stage. The task is the last input to algorithm. The task usually depends on the product or object that will be manipulated by the robot. From the task, a set of the more dominant task points are extracted.

An optimization algorithm is the core of TBCO. This part of the algorithm reads the inputs and finds the optimized kinematic configuration. In this research, Genetic Algorithms are used because of their high efficiency in searching highly nonlinear discrete or continuous spaces. In the GA block, an initial population of the Kinematic Configurations is randomly generated. Each of the individuals of this population represents a kinematic configuration. According to their fitness value the best ones are selected to form a parent pool. To calculate the fitness value for each of the KCs, the Inverse Kinematics and Forward Kinematics should be solved. After undergoing the Crossover and Mutation operators a new generation of kinematic configurations is formed. This process will continue until a termination criterion has been met.

Finally, a decoding stage will convert the output of the optimization algorithm to a kinematic configuration. The output of this stage can include DOF, joint and link types, assembly sequence, and relative orientation of joints.

In the following sections, each of the blocks of the proposed TCO algorithm will be explained in more detail.

1.1 Module Inventory Encoding and Kinematic Configuration Representation

Figure 3 shows the module inventory used in the proposed algorithm. As can be seen, all the links and joints are symmetrical along the axis of the mechanical ports. Therefore, a joint can be connected to a link in four different orientations.

Figure 3) Module Inventory – a) Rotational Joint Module b) Pivotal Joint Module c) Perpendicular-Rotational Joint Module d) Link Modules in three different sizes

To represent a KC assembled from the module inventory, a representation called Kinematic Matrix Representation (KMR) is proposed. Eq.(1.7) shows the KMR of an n DOF robot.

$$R = \begin{pmatrix} 0 & 0 & L_0 \\ \phi_1 & M_1 & L_1 \\ \phi_2 & M_2 & L_2 \\ & \vdots & \\ \phi_n & M_n & L_n \end{pmatrix} \tag{1.7}$$

In KMR, the elements of the first column (ϕ_i) represent the orientation of each joint module relative to the previous one. This orientaion is measured along the axis perpendicular to each mechanical port. In the second column, the joint module types (M_i) are stored. The third column represents the link module types (L_i). The values that each of these variables can store are shown in Table 1. The KMR matrix has 3 column and $n+1$ rows, where n is the DOF of the robot. The first row represents the first link of the robot. This link is perpendicular to the ground and connects the first joint to the base of the robot. Last element of the matrix, at row n+1 and column 3, is the last link which can store a value corresponding to a tool or a spherical wrist.

Table 1) Eligible Values for the variables in KMR

Variable	Value	Meaning
ϕ_i	0	0 Radians
	1	$\pi/2$ Radians
	2	π Radians
	3	$3\pi/2$ Radians

Variable	Value	Module
L_i	0	No Links
	1	Small Link
	2	Medium Link
	3	Large Link
M_i	0	Rotational Joint
	1	Pivotal Joint
	2	Perpendicular-rotational

1.2 Optimization Criteria

As the optimization criteria Relative Manipulability (M_r) is used (Kim 1992). M_r is defined as in Eq.(1.8). Where m and n are the number of rows in the Jacobian Matrix (the dimension of the task space) and the number of joint respectively. a_i and d_i are the link length and joint offset of the ith joint. Larger values of Manipulability in robots are preferable.

$$M_r = \frac{\sqrt[m]{\det(J.J^T)}}{f_M}$$
$$f_M = (\sum_{i=1}^{n} \sqrt{a_i^2 + d_i^2})^2 \qquad (1.8)$$

1.3 Genetic Algorithm and GA operators

1.3.1 The Algorithm

A Genetic Algorithm, through Selection, Cross-over and Mutation operators, finds the individuals (in our case KCs) that have the best fitness values and combines them to produce idividuals that offer better fitness values than their parents. This process continues until the population converges around the individual that have the best fitness value (Goldberg 1989). In our application, a Microbial GA (Harvey 1994) (Harvey 2001) which has been modified for TBCO was implemented. In the original Microbial GA, Tournament Selection is used. In Tournament Selection two individuals are randomly picked from the GA population and after comparison the one with the better fitness value is selected as the winner. In our modified microbial GA, to find the two parents, two sets of Tournament Selections are performed in parallel. The losers of the two Tournaments will be replaced by the offsprings of the crossover of the winners.

To estimate the potential of individuals, a fitness value should be allocated to them. In TBCO, the fitness value is evaluated by f_{rch} and f_{op} . To fit the TBCO problem into the GA structure, the optimization problem of Eq.(1.6) can be reformatted as in Eq.(1.9). In the new equation, the goal is finding an *R* which is able to minimize both f_{rch} and f_{op} .

$$\min_{R \in \mathrm{K}} f_{rch}, f_{op} \qquad (1.9)$$

In most of the existing TBCO algorithms, a weighted sum of f_{rch} and f_{op} is used as the fitness value. But it should be noted that f_{op} becomes of importance only if a KC can satisfy the reachability requirements. Hence the necessity of a selection scheme which considers the priority of Reachability over the optimization criteria arises. Figure 4 shows the Pseudo code of the proposed Selection Scheme.

```
if |f_rch^1 - f_rch^2| ≤ κ
    if f_op^1 ≤ f_op^2  ⇒  Individual 1 is the winner
    else                ⇒  Individual 2 is the winner
elseif ( |f_rch^1 - f_rch^2| ≤ φ & rand ≤ ρ )     (φ〉κ)
    if f_op^1 ≤ f_op^2  ⇒  Individual 1 is the winner
    else                ⇒  Individual 2 is the winner
else
    if f_rch^1 ≤ f_rch^2  ⇒  Individual 1 is the winner
    else                  ⇒  Individual 2 is the winner
```

Figure 4) Pseudo code of the proposed Selection Scheme

Where κ, φ and ρ are constants. According to the scheme, when two KCs are being compared three cases could happen. If the reachability of the two KCs are very close to each other the one with the better f_{op} is the winner. If the reachabilities are fairly close, the one with better f_{op} with a certain probability of ρ will be the winner. Finally, if the difference between the reachabilities is large the one with the better reachability is selected as the winner. It can be observed that the scheme considers f_{op} only if the KCs have approximately the same f_{rch}.

1.3.2 The GA operators

In the proposed TBCO algorithm, the Cross-over operator is implemented as follows. One element of the kinematic configuration matrix is selected randomly. Within the two matrices undergoing crossover, all elements located after the selected element are swapped. To accomplish the Mutation, one of the elements of the kinematic configuration matrix is selected randomly. This element is replaced with another randomly generated number. The mutated number should comply with the element type it is being stored in. Figure 5 shows an example of applying the operators on the KMR.

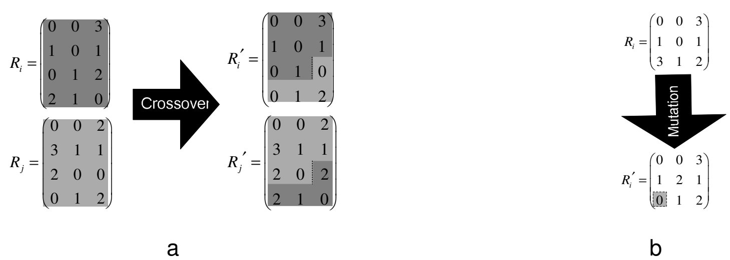


Figure 5) The proposed GA operators: a. Crossover b. Mutation

1.4 Inverse Kinematics

In order to calculate the reachability and most of optimization criteria a solution to the IK problem is required. In most of the existing TBCO algorithms, IK is solved numerically. The numerical IK algorithms converge to the closest solution to their initial value. As a result, only the fitness values that corresponding to only that IK solution will be used in the Tournament Selection. In this case, there always is a chance that a good IK solution of a less suitable KC will be preferred to a worse solution of a desirable KC. To solve this problem, a multi-solution IK solver based on Niching Genetic Algorithms has been developed by the authors (Tabandeh, et al., 2006). The algorithm is developed based on a Niching GA algorithm. Niching GA has the ability to converge on multiple solution regions. In the mentioned algorithm, the output is filtered and the solution regions are detected after being passed through a clustering algorithm. In the present research, the outputs pass through two more blocks, a numerical IK and a selection phase. The numerical IK is used to reduce the error even more. In the selection phase, the best IK solution according to the optimization criteria is chosen to be included in the TBCO algorithm.

Figure 6 shows a block diagram of the IK. The inputs to the algorithm are the robot kinematic configuration and the task in space.

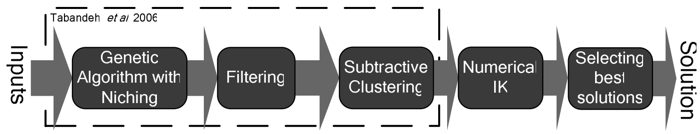


Figure 6) Schematic Diagram of the proposed Multi-solution IK solver

3 Results

The proposed TBCO algorithm was used to seek the most suitable 3DOF configuration for a single point task. Since it was assumed that spherical wrist is used, only position reachability was considered. As was mentioned earlier, Relative Manipulability was used as the optimization criteria. The desired task point is shown in Eq.(1.10).

$$P_{OT} = \begin{pmatrix} -184.8691 \\ 59.911 \\ 129.502 \end{pmatrix} \tag{1.10}$$

$$R_{solution} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 3 & 1 & 3 \\ 1 & 1 & 1 \end{pmatrix} \tag{1.11}$$

The TBCO algorithm was set to run for 200 generations. Figure 7-a shows the progress of the minimum Reachability of each generation with respect to the generation number. As can be seen, the Reachability decreases to 7.5 (3% of the P_{OT}) with the progree of the algorithm. It should be noted that it is possible to gain even lower reachabilities by increasing the maximum generation number. The KMR of the robot that could reach the least reachability in the last generation is shown in Eq.(1.11). Figure 7-b shows the robot corresponding to this KMR. The black sphere close to the end-effector of the robot is the desired task point. The Relative Manipulability of the robot in reaching the task is 0.0866.

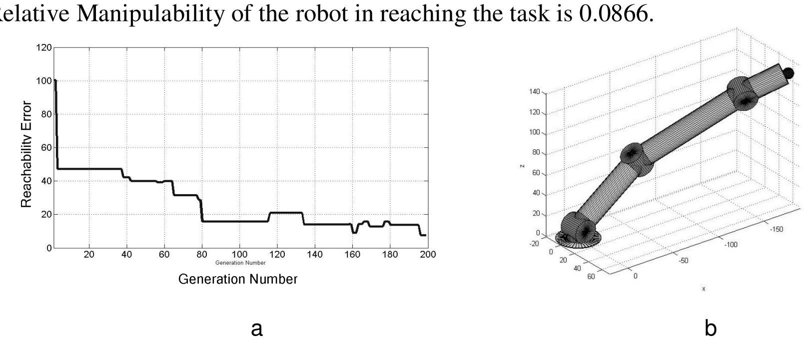


Figure 7) a. Progress of the Reachability with respect to the generation number b. The output of the TBCOalgorithm

4 Conclusions and Future work

A Task-based Configuration Optimization based on a Genetic Algorithm was presented. The algorithm uses a Multi-solution IK solver to calculate the fitness value of Kinematic Configurations. The proposed TBCO algorithm uses a novel Selection scheme. Moreover, modified Crossover and Mutation operators to handle KMRs were explained. The algorithm was used to find a KC capable of performing a certain task while achieving a high relative Manipulability.

To fully utilize the capabilities and deficiencies of the algorithm, implementation of a larger set of optimization criteria is the next phase of the research. Moreover, addition of orientation reachability and expanding the algorithm to search in the space of *n* DOF Kinematics configuration for a larger set of task points could be another focus of the TBCO development.

5 References

Chen IM (1994) Theory and Applications of Modular Reconfigurable Robotic Systems. PhD thesis, Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA, USA

Chen IM, Yan G (2001) Closed-form inverse kinematics solver for reconfigurable robots. Proc. of the IEEE Int. Conf. on Robotics and Automation, pp.2395–2400

Chen IM, Yang G (1998) An evolutionary algorithm for the reduction of dofs in Modular Reconfigurable Robots. Proceeding of the ASME Dynamic System and Control Division, DSC-64, pp.759–766

Leger C (1999) Automated Synthesis and Optimization of Robot Configurations: An Evolutionary Approach. PhD thesis, The Robotics Institute, Carnegie Mellon University, Pittsburg, Pennsylvania 15213

Kim JO (1992) Task based kinematics of robot manipulators. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213-3890, August 1992.

Chocron O, Bidaud P (1997) Genetic design of 3d modular manipulators. Proceedings of the 1997 IEEE Int. Conf. on Robot. And Auto., Albuquerque, New Mexico

Goldberg D (1989) Genetic Algorithms in search, optimization, and machine laearning. Addison-Wesley

Harvey I (1994) chapter Evolutionary robotics and SAGA: the case for hill crawling and tournament selection, Artificial Life III, Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XVI, pp.299–326. Addison Wesley

Harvey I (2001) Artificial evolution: A continuing SAGA. Proc. of 8th Intl. Symposium on Evolutionary Robotics (ER2001)

Tabandeh S, Clark C, Melek W (2006) A genetic algorithm approach to solve for multiple solutions of inverse kinematics using adaptive niching and clustering. Proceedings of 2006 IEEE World Congress on Computational Intelligence, Vancouver, BC, Canada