Configuration Engine for Modular Parallel Robot Assembly

Jason Lemay and Leila Notash
Department of Mechanical Engineering, Queens University, Kingston, Ontario, Canada

1 Introduction

A modular robotic system is an assemblage of parts that can be restructured to an optimal configuration depending on changes in the task, environment or operating parameters. These systems are typically composed of a few standard modular units, with a common interface. Parallel robots, as their name implies, can employ several legs to connect the base to the end effector. The excellent force to weight and stiffness capabilities of a parallel robot allows for greater flexibility in the modular components.

A configuration engine is a computer program that allows modular devices to be optimally assembled for a specified task. The mechanical flexibility increases the possible applications, and allows for different robot structures to be examined. The configuration engine can also determine what modules and module criteria are actually required for the optimal configurations.

The research on modular robot systems has typically been conducted on the optimization of the configuration with an unalterable set of modules. Therefore the criteria required of the modules to produce the optimal configurations, have yet to be examined. Once an engine has been developed, the modules can be easily varied to track the effect on the population, hence determining the required module criteria.

This paper examines the theory required for a modular parallel robot configuration engine. Possible evaluation criteria are identified, as well as calculation methods. An optimization method is proposed to be combined with the calculation methods for these criteria to form the main program structure. Methods for the program implementations are discussed, as well as the possible directions of future work.

2 Optimization Criteria

Four possible traits for optimization are being examined for this configuration engine: workspace, dexterity, weight and size, and accuracy error.

Workspace: The possible workspace of the mechanism is of principal importance for the configuration process. Without the ability to resolve the workspace it is impossible to say if the robot can complete any tasks. The general workspace analysis employed for this configuration engine, consists of determining the distance each leg is required to span for each task point, using a loop closure method [1]. Once this is accomplished the individual legs are analyzed to determine the maximum and minimum lengths they could reach, using a multivariable linear search over the leg joints. The difference between the required distance and the reachable distance is the evaluated number passed to the objective function for the engine.

The workspace evaluation (WE) criteria is defined in equation (1), where L is the length of the leg. The WE is further divided into negative (WEO) and positive components (WEI), for points outside and within the workspace respectively.

$$WE = \min\left(L_{\max} - L_{required}, L_{required} - L_{\min}\right) \tag{1}$$

Dexterity: Dexterity, another important criterion of the robot configuration process, is a measure of the ability of the system to affect the environment at a point. Many methods use the dexterity criterion in the configuration optimization since it can also be used to examine the singularities for a given trajectory [2]. For the dexterity analysis, the Jacobian of the parallel manipulator must be found, and the eigenvalues determined using a singular value decomposition method [3]. The ratio of the smallest and largest eigenvalues, combined with the smallest eigenvalue can then be taken as a measure of the dexterity and used in the objective function. This gives the second criteria in the form shown in equation (2), where DE is the dexterity evaluation, σ is the eigenvalue and w is the weighting factor associated to the eigenvalue.

$$DE = w_1\sigma_{\min} + w_2\frac{\sigma_{\min}}{\sigma_{\max}} \tag{2}$$

Weight and Size: The weight and size criteria can be modeled as simple constraints for the optimization routine. The weight and size model can be taken as the summation of weight and size values associated with the module types used. The weight and size can then be combined to provide the overall weight and size evaluation (WSE), as shown in equation (3), where w_j and s_{ij} indicate the weight and size of the modules, and j indicates the number of the module.

$$WSE = \sum_j s_j + \sum_j w_j \tag{3}$$

Accuracy Error: The accuracy error of the end effector can also be a criterion for the design of modular systems. The error will be formulated using a worst case scenario for the individual legs and a stochastic model for the combination of those legs. The error will be added for the serial leg component and averaged for the total number of legs. Given a maximum error value from the user, a multivariable linear optimization routine will deduce the largest possible tolerances of the modules. The evaluated number will therefore be the tolerance cost evaluation (TCE), and will be calculated as shown in equation (4), where T_i is the joint tolerances joint i and N is the number of legs of the device.

$$TCE = \frac{\max\left[\sum(T_i)\right]}{N} \tag{4}$$

3 Program Methodology

The program can be separated into six segments, the input, population initialization, objective function generation, population evaluation, optimization algorithm and output. A flow chart of the program is presented in Figure 1.

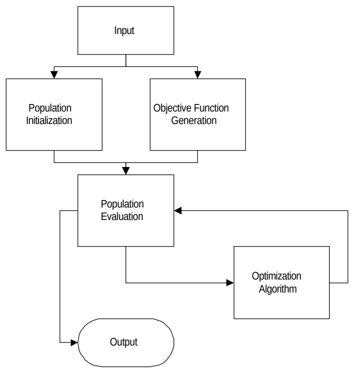

Figure 1. Program Flow Chart.

The configuration engine requires the task definition, operating conditions, available modules, optimization criteria and optimization algorithm variables as input. The task is currently defined as a series of points in space that the robot must reach and operate at.

The population initialization section contains the algorithm that creates the first population and determines the starting point for the optimization routine. The initial parallel configurations are generated randomly based on the degree of freedom (DOF) and available modules.

The objective function section controls the criteria that are being considered and their weighting factors. The user can to choose the criteria to be optimized. The weighting factors are required to allow for the scaling of the different criteria evaluations and are dependent on the DOF of the device. The objective function (OF) can be seen in equation (5) where the $F_{i,}$ i = 1 to 5, are the weighting factors for the different criteria.

$$OF = F_1 * WEO + F_2 * WEI + F_3 * DE + F_4 * WSE + F_5 * TCE \quad (5)$$

The population evaluation is the control component of the program. The population evaluation evaluates the individual criteria and combines them using the objective function. The total and highest evaluation score can then be determined and used to process several decisions, such as the redefinition of the optimization algorithm variables or used as the stop criteria for the program.

The optimization algorithm creates a new population generation using a variation of the Adaptive Simulated Annealing and Genetic Algorithm (ASAGA) process [4]. The ASAGA process is a genetic algorithm (GA) with a simulated annealing (SA) mutation component. The GA applies biological evolution laws to a population to form a new, evolved, population. There are three laws that are applied by the GA, reproduction, crossover and mutation. For this work the mutation process has been replaced by a SA method to increase the speed and reliability of the algorithm. The optimization variables, which control the rates of reproduction, crossover and mutation, are redefined by the engine or changed by the user if the optimization does not improve the total population evaluation score.

The output displays the best configuration available, at each iteration and for the final population, as an array and with its evaluation score. Other displayed factors include the number and type of modules used, failures in task completion and the number of cycles required for the convergence of the optimization algorithm.

4 Results

The configuration engine has been completed for two evaluation criteria, workspace and size and weight. Currently it only has three simple modules available for the configuration, one revolute (R) type and one prismatic (P) type of joint modules and one static type of link module. The engine was successfully tested by giving points known to be within the workspace of a planar parallel robot (Figure 2b). The engine identified two other known parallel structures, one similar to the one in Figure 2a, and one with RRP legs.

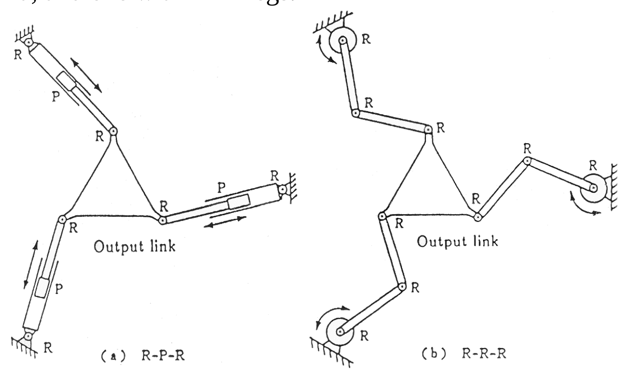

Figure 2. Generated Linkages [5].

5 Future Work

Further criteria components, such as the ones mentioned, will be added time permitting. Other criteria than the ones mentioned here could include the force capabilities, fault tolerance of the robot and path planning for self-reconfiguring robots.

Modified modules will be given to the configuration engine to determine the criteria that the modules require in producing the optimal configurations. In addition, the task definition could be changed to include trajectories as opposed to points. The evaluation criteria methods would have to be altered in relation to the changes in the task definition.

References

[1] J.P. Merlet: "Designing a Parallel Manipulator for a Specific Workspace", Int. J. Robotics Research, Vol. 16, No. 4, pp. 545 - 556 (1997).

[2] W. Zhang and J.P.T. Mo: "Positioning and Orientation in Modular Robot Systems", Proc. IEEE Int. Conf. Control, Vol. 2, pp. 1140-1144 (1994).

[3] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery: "Numerical Recipes in C", Cambridge University Press, New York, 1992.

[4] I.K. Jeong and J.J. Lee: "Adaptive Simulated Annealing Genetic Algorithm for System Identification", Trans IEEE Artificial Intelligence, Vol. 9, No. 5, pp. 523 - 532 (1996).

[5] M. Uchiyama: "Structures and Characteristics of Parallel Manipulators", Int. J. Advanced Robotics, Vol. 8, No. 6, pp. 545 - 557 (1994).