2001

On Modular Design of Field Robotic Systems

Shane Farritor
University of Nebraska - Lincoln, sfarritor@unl.edu

Steven Dubowsky
Massachusetts Institute of Technology

Published in *Autonomous Robots* 10 (2001), pp. 57–65.

On Modular Design of Field Robotic Systems

Shane Farritor

Department of Mechanical Engineering, University of Nebraska–Lincoln, Lincoln, NE 68588 U.S.A.

Steven Dubowsky

Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 U.S.A.

Abstract

Robots are needed to perform important field tasks such as hazardous material clean-up, nuclear site inspection, and space exploration. Unfortunately their use is not widespread due to their long development times and high costs. To make them practical, a modular design approach is proposed. Prefabricated modules are rapidly assembled to give a low-cost system for a specific task.

This paper described the modular design problem for field robots and the application of a hierarchical selection process to solve this problem. Theoretical analysis and an example case study are presented. The theoretical analysis of the modular design problem revealed the large size of the search space. It showed the advantages of approaching the design on various levels.

The hierarchical selection process applies physical rules to reduce the search space to a computationally feasible size and a genetic algorithm performs the final search in a greatly reduced space. This process is based on the observation that simple physically based rules can eliminate large sections of the design space to greatly simplify the search.

The design process is applied to a duct inspection task. Five candidate robots were developed. Two of these robots are evaluated using detailed physical simulation. It is shown that the more obvious solution is not able to complete the task, while the non-obvious asymmetric design develop by the process is successful.

Keywords: modular robotics, modular design, genetic design, field robots, mobile robots

1. Introduction

Robots are needed to perform important field tasks. Despite their advantages they are not widely used, largely due to their long development times and high costs. To be practical, field robot systems should be ready in weeks or months and cost only tens of thousands of dollars. New design approaches are needed.

To make these systems practical, a modular design method is proposed (Farritor et al., 1996). Here, an inventory of prefabricated modules is used to rapidly and cost-effectively produce a robotic system for a specific task. The inventory includes actuated joints, links, end-effectors, and power units. The same inventory can be assembled in different configurations to perform different tasks.

Using an inventory of "standard" modules greatly shortens development times and reduces costs. This paper presents a methodology to determine the best robot assembly for a given task.

2. Background

Previous research on mobile field robots has largely focused on either the development of a specific technology, or a "one-of-a-kind" system.

There has been work in developing modular field robotic systems. Many approaches propose identical modules that can be combined in various ways to produce useful robots with various forms of locomotion and manipulation (Hamlin and Sanderson, 1997; Murata et al., 1998; Kotay et al., 1998; Chirikjian and Pamecha, 1996). These studies do not directly address the task-based configuration selection problem discussed in this paper.

There has been important work on modular industrial manipulators. These studies have dealt with the design (Ambrose and Tesar, 1992; Paredis et al., 1996), kinematic modeling (Kelmar and Khosla, 1990), and dynamic modeling (Chen and Yang, 1997).

Task-based design of modular serial manipulators has been studied. One computationally intensive method simulates each manipulator and uses a modified genetic algorithm (GA) (Paredis, 1996). Another method uses a GA, but limits the design to one kinematic shape (Chen and Burdick, 1995).

Configuration design of field systems differs from industrial manipulator design. The diversity in topology and need for mobility as well as manipulation prohibits the direct application of the above work. New methods are required.

3. The Modular Design Problem

The goal of the modular design problem is to select the best assembly of modules for a given task. Here, this is viewed as a search of a design space.

The basic assumption of a modular approach is that useful designs can be created for a reasonable amount of tasks with a reasonably sized inventory. Note that this approach sacrifices optimality compared to a design that is independently created for a specific task. Instead a sufficient, cost-effective, rapid design is created.

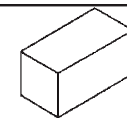
3.1. Conventional versus Modular Design

In important ways, the design of a modular system is simpler than a conventional system. Conventional design variables are in general continuous, and the number of possible solutions is infinite. The modular design space is discrete with a finite size. Theoretically, this space could be enumerated and every possible design evaluated. In the following section it is shown that the size of the space grows very rapidly with the number of available modules. For any real problem an exhaustive search is not practical.

3.2. The Modular Robot Design Space

Consider the simple inventory shown Table 1. The number of possible assemblies that can be created using a given inventory can be computed with a set of robot assembly rules, for example:

Table 1. A simple inventory.

Name	Quantity	
Power/Control Module	$n_\mathrm{p} = 1$ $N_\mathrm{ports} = 14$	
Joint	$n_\mathrm{joints} = 4$	
Foot	$N_\mathrm{feet} = 2$	

1) Assemblies must have a power/control module.
2) Modules are assembled in serial chains called limbs and are attached to ports on the power/control module.
3) All limbs must terminate in an end effector.
4) All modules do not need to be used.

In Table 1, n_p, n_joints and n_feet are the number of power/control modules, joints and feet in the inventory respectively, N_ports is the number of locations that limbs can be attached, called ports.

The number of possible designs, D, that can be made from this inventory is the product of two factors: the number of possible limbs and the number of configurations for these limbs.

$$D = D_\mathrm{ports} \times D_\mathrm{limbs} \tag{1}$$

$$D_\mathrm{ports} = \frac{N_\mathrm{ports}!}{i!(N_\mathrm{ports} - i)!} \tag{2}$$

$$D_\mathrm{limbs} = \frac{(j + i - 1)!}{j!(i - 1)!} \tag{3}$$

Where D_limbs, is the number of possible limbs, D_ports is the number of ways these limbs can be attached, j is the number of joints used, and i is the number of limbs. Since each limb must terminate in an end effector, the number of limbs is equal to the number of end effectors used.

This product, D, is summed over i (where i varies from 0 to the number of end effectors, n_feet.) and j (where j varies from 0 to the number of joints, n_joints) to determine the total number of possible designs (Farritor, 1998-1).

$$D = \sum_{i=1}^{n_\mathrm{feet}} \left[\left(\frac{N_\mathrm{ports}!}{i!(N_\mathrm{ports} - i)!} \right) \left(\sum_{j=0}^{n_\mathrm{joints}} \frac{(j + i - 1)!}{j!(i - 1)!} \right) \right] \tag{4}$$

The size of that search space varies with the number of joints and end effectors. The search space for the small inventory of Table 1 contains 2800 possible robots. The modular design space, even for such a simple inventory, grows rapidly with the number available modules. A more realistic inventory with 14 joints, 8 links and 7 end-effectors can produce over 10^{20} robots (Farritor, 1998).

It is important to observe that the inventory could consist of limbs, or *higher-level modules*, instead of individual joints and links. Such a design space is dramatically smaller. Again, the total number of assemblies is the product of the number of limbs that can be created, and where these limbs can be placed on the power module. With higher-level modules the number of possible limbs is reduced.

Consider a *higher-level* inventory with one power/control module as in Table 1 and two types of "limb"modules (six of each). The number of designs that can be produced with this *higher-level* inventory is $3{:}34 \times 10^7$, compared to the 10^{20} designs from an *low-level* inventory (Farritor, 1998-1). However, this is still a large number for such a simple inventory (two high-level modules) too large to be exhaustively searched. With this *high-level* inventory it is possible to construct a robot with up to 12 limbs. If robot assemblies are limited to 7 limbs (a realistic design), there are just over 700,000 possible designs as compared to the 10^{20} robots of the *low-level* inventory, a reduction of 10^{15}. These observations are utilized in the hierarchical design approach.

Requiring the robot to be symmetric would also reduce the size of the design space (by reducing both N_{ports} and the number of modules). However, this places too great a restriction on the final design. It will be shown in Section 5 that an asymmetric, non-obvious robot may be the best solution.

4. The Hierarchical Design Approach

A Hierarchical Selection process is proposed to search the modular design space for the best assembly to accomplish a given task. The process consists of tests and filters used at various levels of the process. It eliminates entire sub-trees of solutions from further consideration by exploiting the physical nature of the system and the task. This reduces the search space to a computationally feasible size. Then a genetic algorithm is applied to perform the final search in a greatly reduced search space.

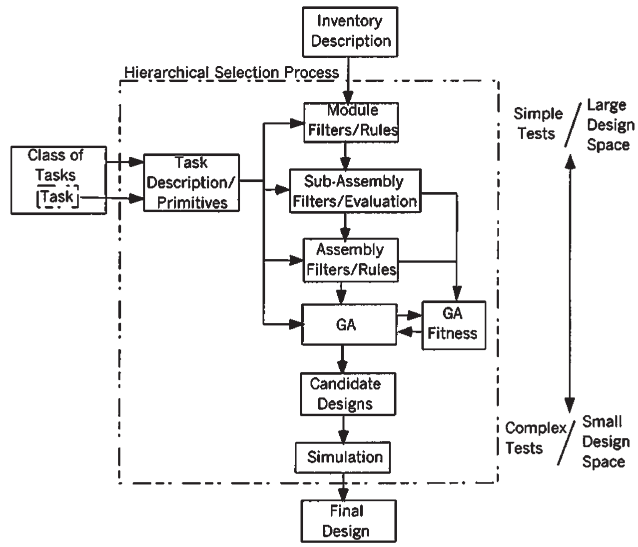


Figure 1. The modular robot design structure.

This process is based on the observation that simple physically based rules can eliminate large sections of the design space to greatly simplify the search (Farritor et al., 1996). The method applies the simplest and computationally inexpensive tests first to prune the search space and quickly converge on a smaller set of candidate solutions. Only the successful candidates need to be considered by more computationally intensive tests.

The selection process is outlined in Fig. 1. First the design problem is considered at the module level, then the subassembly level, and finally at the assembly level where the genetic algorithm is used to search the greatly reduced space to produce candidate designs. Then more complex tests are used to select a final design.

The selection process is based on some simple assumptions. First, it is assumed that computationally simple tests can help distinguish between "good" and "bad" designs. Second, it is assumed that a robot can be designed without precise knowledge of how it will execute the task. However, the final stages of the design process may require some iteration between design and planning. With these assumptions there is no guarantee of optimality, instead the process finds a sufficient design.

4.1. Task and Inventory Descriptions

To design a robot a description of the task is required. Here tasks are described by a combination of task primitives that are relevant to a class of tasks being considered. This paper considers inspection robots for pipe

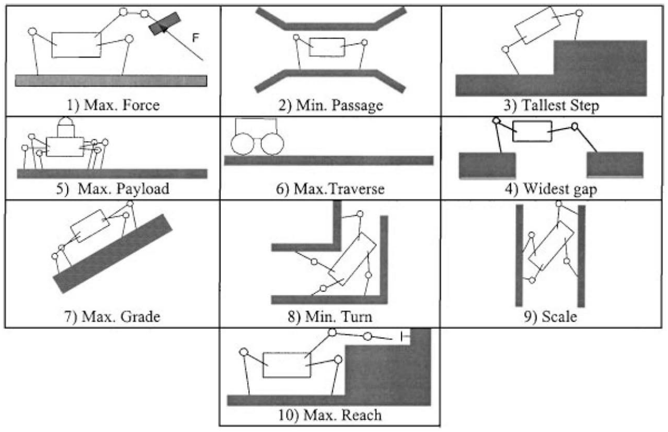


Figure 2. Task primitive inventory.

and duct networks as well as small, enclosed rooms or channels. Such tasks can be found in the telecommunication industry, city infrastructure and large buildings. The task and primitives, shown in Fig. 2, are used to create the tests and filters of the selection process. All tasks in this class of tasks are a combination of these primitives. Table 2 shows a set of simple tests derived from the task primitives. Other constraints can be also added such as the maximum robot cost or weight.

The module inventory is characterized before the design process begins, Table 3. It includes power/control, joint, link, and end effector modules. Robots are constructed following the assembly rules of Section 3.2. All robots contain power modules where serial chains of modules can be attached. The ports provide an energy connection of one of two types, electric or pneumatic. Modules of different energy types are not compatible.

Table 2. Example simple tests.

Task requirement	Example simple test
Max. applied force	F_{endpoint}
Smallest passage	X, Y, Z size
Tallest step	Limb length/strength
Widest gap	Limb length/strength
Max. payload	F_{endpoint} all limbs
Max. traverse	Available energy
Max. grade	limb strength coefficient of friction
Min. turn	X, Y, Z size
Max. reach	Maximum limb length
Scale	limb strength coefficient of friction
Time to complete task	velocity/max. traverse

Joint modules are available with various sizes, strengths and speeds and can be attached in two configurations, corresponding to a 90-degree rotation of their axis. Wheels, feet, and grippers are included. A wheel and a gripper can be used as a foot. The inventory also contains connecting link modules for dimensional changes. This inventory can produce a great diversity of robot configurations.

Sensor modules for obstacle avoidance and navigation are not considered in this paper but could be easily included (Farritor, 1998-1).

4.2. Module and Sub-Assembly Evaluations

The selection process begins by applying module filters derived from the task and inventory descriptions. If a module can be removed early in the design process, it will eliminate a vast number of sub-assemblies and an even larger number of assemblies. Filters at the early stages greatly reduce the size of the design space.

The module filters eliminate modules that are not appropriate to the task. For example, if a robot needs to pass through a small opening, all modules that are larger than this opening are eliminated. Table 4 shows some example module-level filters and tests.

Next, the design is analyzed on the sub-assembly level. Entire sub-assemblies and groups of subassemblies can be eliminated from consideration. For example, sub-assemblies that do not contain joints are not useful. More complex tests can also be applied to sub-assemblies such as the size of the limb workspace, or a sub-assembly Jacobian can be developed to determine parameters such as the maximum applied force, nominal power consumption per unit applied force, or maximum endpoint velocity. The sub-assembly tests used in this paper are shown in Table 5.

The evaluation of sub-assemblies can be viewed as the development of an inventory of high-level components. A high scoring sub-assembly can be thought of as a single component in a higher-level inventory. The reduction in the search space using a higher-level inventory was shown in Section 3.2.

4.3. Assembly-Level Evaluation

Finally, the design process considers a complete robot. Examples of assembly evaluations can be seen in Table 6.

With the design space substantially reduced, but still large, a genetic algorithm (GA) is used to search for the

Table 3. Module inventory.

ID#	Energy type	Quantity	Weight (oz).	Dimension (in.)	Notes
Power/control modules					
001	E	1	48	8 × 4 × 4	14 ports/computation
002	E	1	16	3 × 4 × 4	4 ports/power only
003	E	1	16	3 × 4 × 4	4 ports/power only
004	P	1	60	16 × 8 × 8	16 ports/computation
Joint modules					
101	E	6	1.5	2.25 × 1.5 × 1	42 oz-in stall
102	E	6	3.3	2.25 × 1 × 1	92 oz-in stall
103	E	6	2.8	2.5 × 1.3 × 1.8	200 oz-in stall
105	E	4	2.8	2.5 × 1.3 × 1.8	Non-backdrive 300 oz-in stall
151	P	6	5.5	1 × 3 × 1	200 oz.-in. stall
152	P	6	6.2	1.5 × 4 × 2	325 oz-in stall
153	P	6	8.0	2 × 6 × 3	580 oz-in stall
End effector modules					
301		8	.25	1 × 1 × 1	Rubber foot
302	E	8	.65	1 × 1 × 1	Magnetic foot 16 oz. break-away force
303		8	.25	1 × 1 × 1	Suction cup 10 oz. break-away force
304	E	6	3.5	2.5 × 2.5 × 1	Wheel 150 oz.-in. stall/Dwheel = 2
305	E	4	5.	2.5 × 2.5 × 4	Track 150 oz.-in. stall
306	E	1	1.5	1.5 × 1 × 2	Gripper/.6 lbf grip
307		1	8.	2 × 2 × 3	Gripper/6 lbf grip
Link modules					
201		12	.5	1 × 1 × 1	
202		12	1.0	1.5 × 1 × 1	
203		12	2.0	2 × 1 × 1	

Table 4. Sample module filters and tests.

External module filters
- Module weight < W_{max}
- Module cost < C_{max}

Geometric module filters
- Module size < l_{max}
- Gripper span > d_{object}

Function module filters
- Gripper force > W_{object}

Module energy domain filters
- Discard modules w/o power sources
- Discard power sources w/o modules

Table 5. Example sub-assembly filters and tests.

Filters
- Weight < W_{max}
- Cost < C_{max}
- Must terminate with an end effector
- Maximum of 3 joints per limb

Kinematic analysis
- x reach
- y reach
- z reach
- DOF
- $F_{max} = [F_x ; F_y ; F_z]$

Power analysis
- Average power consumed

Mobility analysis
- Power/(velocity × weight)

candidate designs. A GA is used because of its effectiveness in searching large diverse spaces. The GA represents assemblies with a tree structure, or chromosome. The GA begins with a number of random robot assemblies, called a generation. The algorithm combines attributes (modules in this case) from one assembly with those of another, creating a new generation of robots. This process is called crossover. Robots are chosen for crossover to make "better" robots more likely to appear in the next generation. The algorithm may also add new characteristics (modules) that were not present in the previous generation. This process is called mutation. This is a fairly standard application of a steady state GA (Goldberg, 1989).

The genetic algorithm evaluates robots using a fitness function. The tests and filters shown in Table 6 are

Table 6. Example assembly filters and tests.

Filters
Cost < C_{max}
Weight < W_{max}
Kinematic analysis
Static stability
x reach
y reach
z reach
F_{max}
Power analysis
Average power
Peek power
Operating time
Power for mobility
Mobility analysis
DOF
Velocity
Power/(velocity × weight)
Max distance

used to produce a fitness value that estimates a robot's performance. Section 5 contains a further explanation of the fitness functions.

Table 7. Duct task test parameters.

Task requirement	Simple test	Quantity
Smallest passage	Width–length	11″–11″
Tallest step	Limb length–strength	2″–$F_{z\text{-max}}$
Widest gap	Limb length–strength	8″–$F_{z\text{-max}}$
Max. traverse	Energy available	min(t_i)
Max. grade	Limb strength	Max($F_{z\text{-max}}$)
	coef. of friction	μ > tan.30)
Min. turn	y size–x size	11″–11″
Max. reach	Max. limb length	8″
Time	Velocity	Max/velocity

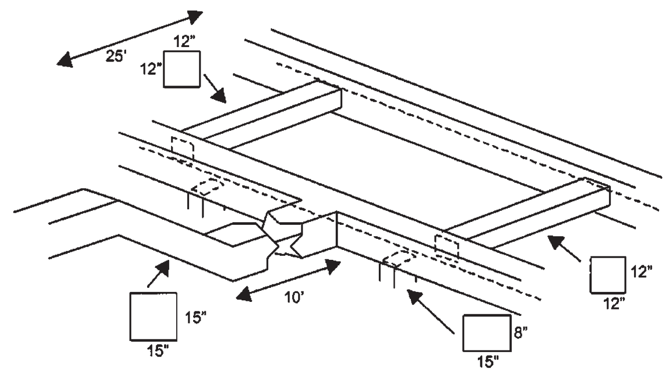


Figure 3. Duct work inspection task.

5. A Duct Work Inspection Task

The modular design process is demonstrated on a duct work inspection task. Often, difficult to access ducts need to be inspected in many industrial and municipal areas. Candidate designs are presented and tested using detailed simulation. Then, a final design is selected.

5.1. *Problem Definition*

An inspection of the duct is needed requiring the robot to travel throughout the duct network, see Fig. 3. The robot will be inserted at the left of the figure. It will need to travel down a 30° slope and make 90° turns in a 12″ duct. It will also need to climb a step of 2″ and cross an 8″ gap. The task is described using the task primitives of Fig. 2 and is summarized in Table 7.

The Hierarchical selection process begins on the module-level (Table 4). As an example, the long sections of ducts, along with turns in the network, prohibit the use of a tethered module. The pneumatic power supply and therefore other modules that require pneumatic power were eliminated.

Next, the design problem was considered on the sub-assembly level (Table 5). The 8″ gap to be crossed causes the sub-assembly tests to favor longer limbs. Also, because of the long distances, sub-assemblies (and designs) must be power efficient.

Finally, the design problem was considered on the assembly level (Table 6) and the genetic algorithm searched the reduced design space.

The algorithm assigns a fitness to a configuration by making estimates of the robot's performance characteristics including cost, weight, static stability, climbing ability, average velocity and average power consumption. The fitness function uses the simple form shown in Eq. (5). Where p_i is a number between 0 and 1 that estimates robot performance and w_i is a weighting factor.

$$f = \sum_{i=1}^{n} w_i p_i \tag{5}$$

An example performance characteristic is the robot's ability to cross the required 8″ gap. It is estimated using the maximum robot length. For instance, a robot with 6″ span will not be capable of crossing the required gap. A

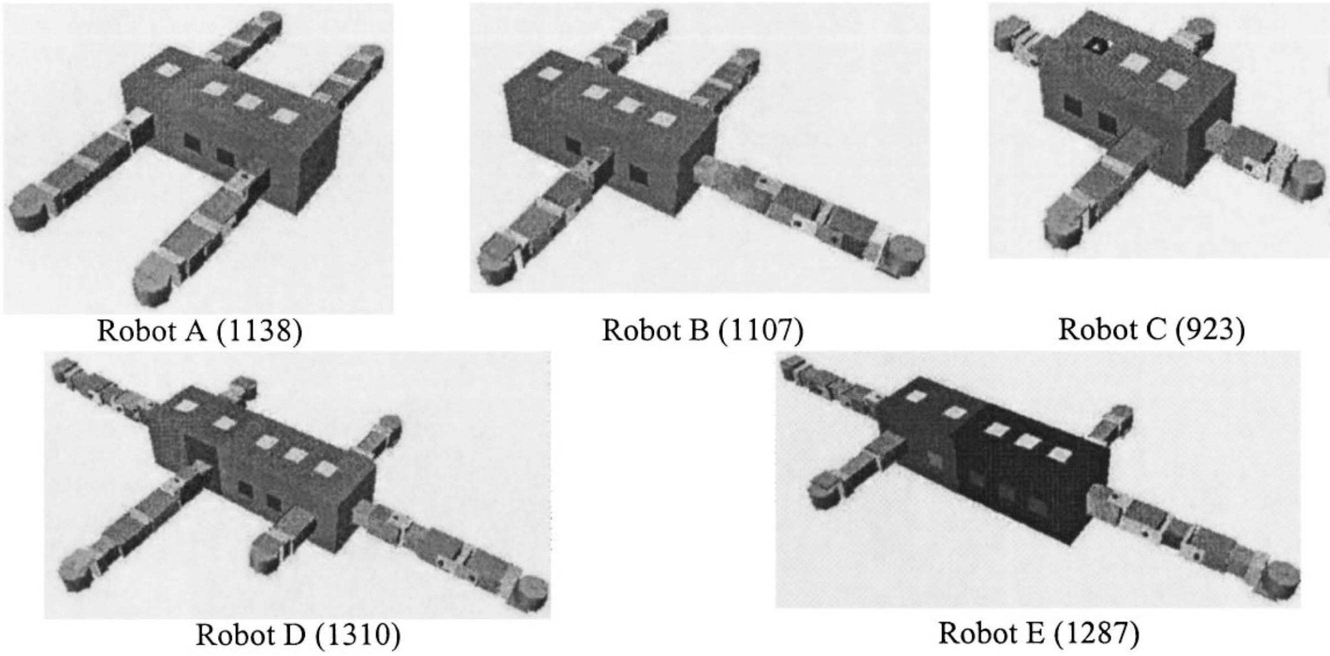


Figure 4. Candidate robot designs.

second performance characteristic estimates the robot's velocity by developing a Jacobian for each limb and averaging maximum end-effector velocities.

This GA search used a population of 100 individuals and approximately 3000 generations were required for convergence. Crossover was performed at a 60% rate and a mutation at a 2% rate. The search was completed in approximately 140 minutes using a Sparc Classic Sun workstation.

The GA devloped five candidate designs, shown in Fig. 4 with their relative fitness scores. The selection process favored robots with wheels because of the long distances and relatively simple climbing requirements. Also, the requirement to cross an 8″ gap caused the selection process to favor long robots.

Many of the robots have similar limb configurations. For instance, the kinematic configuration seen on the front and rear of Robot E, can also be found on robots A, B and D. This kinematic configuration was favored during the sub-assembly evaluation because it is a long limb that can support a large vertical force with little power.

Robot A is the most obvious design. It has four limbs and uses an additional power supply module #002 to increase its span and operating time. Robot B is somewhat similar to robot A in that it also has four legs and uses the same power supply. Robot C only uses the power/control module #001. It has few joints and therefore can operate for a long time and travel long distances. However, it is not highly mobile. Robots D and E are also similar. Each use a #002 and #003 power module to increase operating time and span. Each has long limbs on the front and rear to increase the robot's span.

Because of the similarity between Robots A and B and between D and E, Robots A and E were further evaluated. Finally, this greatly reduced design space (two robots) is evaluated using a computer simulation. The robot was required to travel down the slope, turn right and cross the 8″ gap. Then climb the 2″ step into the narrow duct at the right of Fig. 3.

The simulation considered physical constraints such as limb interference, geometric limitations, static stability, actuator saturation, and power consumption. Because of the relatively slow motion of the robots, dynamics were not considered.

Power consumption is one of the key performance factors considered by the simulation so it is explained here as an example. It is assumed the actuators are the dominant power consuming elements and power requirements are proportional to motor torques (Dubowsky et al., 1995). To estimate the motor joint torques the foot reaction forces are found. When the robot has four legs in contact with the ground, the problem is statically indeterminate so compliance is introduced at each contact point, see Fig. 5.

It is also assumed that the surface is relatively level so slip and tangential forces are not relevant and that the robot elements are rigid. A kinematic analysis determines the configuration of the robot at each instance.

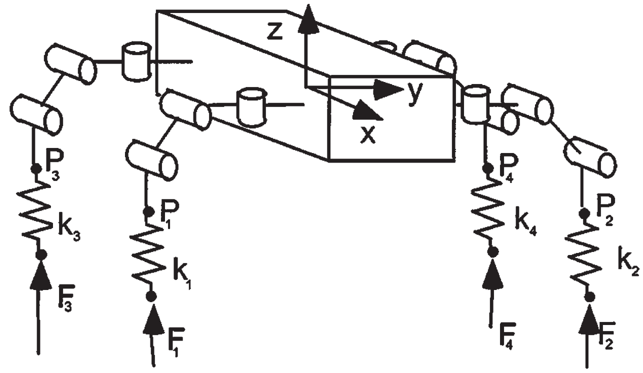


Figure 5. Calculation of reaction forces.

Then static equilibrium yields:

$$\sum F_z = 0 \;:\; F_1 + F_2 + F_3 + F_4 - W = 0 \tag{6}$$

$$\sum M_x = 0 \;:\; -F_1 y_1 + F_2 y_2 - F_3 y_3 + F_4 y_4 = 0 \tag{7}$$

$$\sum M_y = 0 \;:\; -F_1 x_1 - F_2 x_2 + F_3 x_3 + F_4 x_4 = 0 \tag{8}$$

$$F_n = k_n d_n \tag{9}$$

Where d_n is the compression of spring n, W is the robot weight, and x_n, y_n and z_n are the foot position defined with respect to the robot center of mass. Since the robot is assumed to be rigid a fourth equation relating d_1 to d_4 is written. For instance, if the robot is on a flat surface all feet must lie in a plane, Eq. (10).

$$A(x_4 - x_1) + B(y_4 - y_1) + C(z_4 - z_1) = 0 \tag{10}$$

Where A, B, and C are the parameters of a plane defined by the foot positions P_1, P_2, and P_3. This leaves four equations and four unknowns.

With knowledge of the foot reaction forces, the joint torques can then be estimated using the limb Jacobian. These torques are then used to estimate power consumption and to check actuator saturation.

The robots were tested executing the task using an action plan developed specifically for the robot and task. The biggest challenge was crossing of the 8″ gap found in the middle of the lowermost duct. Robot A was unable to complete the task because the arrangement of its legs did not allow it to reach across the gap while maintaining stability, see Fig. 6. However, the asymmetry and long span of Robot E made it successful. These results show that the asymmetry of Robot E is a good design for this task, while the more obvious solution (Robot A) was not.

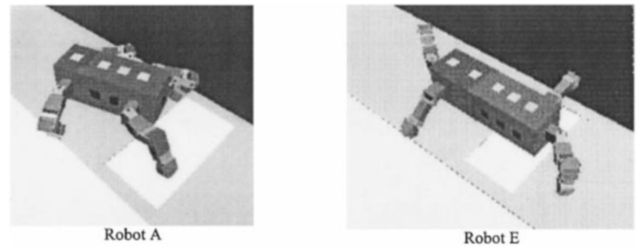


Figure 6. Simulation results.

6. Summary and Conclusions

This paper described the modular design problem for field robots and the application of a hierarchical selection process to solve this problem. Theoretical analysis and an example case study were presented.

The theoretical analysis of the modular design problem revealed the large size of the search space. It showed the advantages of approaching the design on various levels.

The design process was applied to a duct inspection task. Five candidate robots were developed. Two of these robots were further evaluated using detailed physical simulation. It was shown that the more obvious solution was not able to complete the task, while the non-obvious asymmetric design developed by the process was successful.

Acknowledgments

NASA-JPL supports this work. The authors would like to acknowledge the support of Dr. Paul Schenker and his research team at JPL.

References

Ambrose, R. and Tesar, D. 1992. Modular robot connection design. In *ASME DTM*.

Chen, I. and Burdick, J. 1995. Determining task optimal robot assembly configurations. In *IEEE Intl. Conf. on Robotics and Automation*, pp. 132–137.

Chen, I. and Yang, G. 1997. Automatic generation of dynamics for modular robots with hybrid geometry. In *IEEE ICRA*, pp. 2288– 2293.

Chirikjian, G. and Pamecha, A. 1996. A useful metric for modular robot motion planning. In *Proceedings of 1996 IEEE ICRA*, Minneapolis, MN.

Dubowsky, S., Moore, C., and Sunada, C. 1995. On the design and task planning of power efficient field robotic systems. In *Proc. 6th ANS Robotic and Remote Systems*.

Farritor, S. 1998. On modular design and planning for field robotic systems. Ph.D. Thesis, Dept. of M.E., MIT, Cambridge, MA.

Farritor, S., Dubowsky, S., and Rutman, N. 1996. On the rapid design of field robotic systems. In *ASME Conf. on Design Theory and Methodology*.

Farritor, S., Hacot, H., and Dubowsky, S. 1998. Physics-based planning for planetary exploration. In *IEEE Conference on Robotics and Automation*.

Goldberg. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addision-Wesley.

Hamlin, G. and Sanderson, A. 1998. *Tetrobot: A Modular Approach to Reconfigurable Paralles Robotics*. Kluwer Acad. Pub.: Newton, MA.

Kelmar, L. and Khosla, P. 1990. Automatic generation of forward and inverse kinematics for a reconfigurable modular manipulator system. *Journal of Robotic Systems*, 7(4):599–619.

Kotay K., Rus D., Vona M., and McGray C. 1998. The selfreconfiguring robotic molecule. In *Proc. IEEE Intl. Conf. On Robotics and Automation*.

Murata, S., Kurokawa, H., Yoshida, E., Tomita, K., and Kokaji, 1998. A 3-D self-reconfigurable structure. In *Proc. IEEE Intl. Conf. On Robotics and Automation*.

Paredis, C. 1996. An agent-based approach to the design of rapidly deployable fault tolerant manipulators. Ph.D. Thesis, Dept. of Elec. and Comp. Eng., CMU, Pittsburgh, PA.

Paredis, C.J.J., Brown, H.B., and Khosla, P.K. 1996. A rapidly deployable manipulator system. In *IEEE ICRA*, April 22–28, pp. 1434–1439.

Shane Farritor received his Bachelor's degree from the University of Nebraska-Lincoln in 1992, and his M.S. and Ph.D. degrees from the Massachusetts Institute of Technology in 1994 and 1998. He is currently an Assistant Professor of Mechanical Engineering at the University of Nebraska. Dr. Farritor's research interests include modular design, mechatronics, and industrial and mobile robotics. He is currently serves on the AIAA Space Automation and Robotics Technical Committee.

Steven Dubowsky received his Bachelor's degree from Rensselaer Polytechnic Institute of Troy, New York in 1963, and his M.S. and Sc.D. degrees from Columbia, University in 1964 and 1971. He is currently a Professor of Mechanical Engineering at M.I.T. He has been a Professor of Engineering and Applied Science at the University of California, Los Angeles, a Visiting Professor at Cambridge University, Cambridge, England, and Visiting Professor at the California Institute of Technology. During the period from 1963 to 1971, he was employed by the Perkin-Elmer Corporation, the General Dynamics Corporation, and the American Electric Power Service Corporation. Dr. Dubowsky's research has included the development of modeling techniques for manipulator flexibility and the development of optimal and self-learning adaptive control procedures for rigid and flexible robotic manipulators. He has authored or coauthored nearly 100 papers in the area of the dynamics, control and design of high performance mechanical and electromechanical systems. Professor Dubowsky is a registered Professional Engineer in the State of California and has served as an advisor to the National Science Foundation, the National Academy of Science/Engineering, the Department of Energy, and the U.S. Army. He has been elected a fellow of the ASME and is a member of Sigma Xi, and Tau Beta Pi.