On Modular Design of Field Robotic Systems

Shane Farritor
Department of Mechanical Engineering
University of Nebraska
Lincoln, NE 68588

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.

In this paper, a methodology is developed to determine how a modular field robot should be assembled for a given task. The assumptions of the approach are stated and explored within the context of two example tasks.

1 Introduction

Robots are needed to perform important field tasks. Potential applications include hazardous material clean-up, nuclear site inspection, bomb disposal, space exploration and infrastructure inspection (Hayati *et al.*, 1997, Matthies *et al.*, 1995, Schenker *et al.*, 1997, Weisman, 1992). Despite their anticipated advantages they are not widely used. This is largely due to their long development times and high costs; they can require years to develop and cost hundreds of thousands of dollars. To be practical, field robot systems should be ready in weeks or months and cost only tens of thousands of dollars. Clearly, new design approaches are required.

To make these systems practical, a modular design method is proposed (Farritor *et al.*, 1996; Cole, 1995; Rutman, 1995). Here, an inventory of prefabricated modules is used to rapidly and cost-effectively produce a robotic system for a specific task, Figure 1. The inventory includes actuated joints, links, end-effectors, and power units. The same inventory can be assembled in different configurations to perform different tasks. Software action modules are also assembled to produce an execution plan for a given robot assembly and its task (Farritor *et al.*, 1998-2).

Using an inventory of “standard” modules would greatly shorten development times and give substantial cost savings. This paper presents a methodology to determine how a modular field robot should be assembled for a given task.

2 Background

Previous research on mobile field robotic systems has explored different concepts for mobility such as walking, climbing, rolling and crawling (Hirose, 1993; Weisman, 1992; Nagakubo and Hirose, 1994; Pratt *et al.*, 1997; Schenker *et al.*, 1997; Hayati *et al.*, 1997; Pamecha *et al.*, 1996). These works largely focus on either the development of a specific technology, or on a specific “one-of-a-kind” system.

Recently, there has been important work on industrial manipulators constructed of modular components. These studies have dealt with the mechanical design (Tesar and Butler, 1989; Ambrose and Tesar, 1992; Cohen *et al.*, 1992; Paredis et al, 1996), kinematic modeling (Benhabib *et al.* 1989; Kelmar and Khosla, 1990), and dynamic modeling (Chen and Yang, 1997) of such systems.

Configuration selection for modular industrial manipulators based on task requirements has been explored. A computationally intensive method that simulates the performance of a modular manipulator performing a task and uses a search technique based on a modified genetic algorithm has been proposed (Paredis and Khosla, 1993; Paredis, 1996). A second method uses a genetic algorithm, but limits the search to one kinematic configuration (Chen and Burdick, 1995). The use of genetic algorithms in design is a growing area of research. They have been used to design non-modular robots (Kim and Khosla, 1993; Chedmail and Ramstein, 1996) and in other design applications (Wallace, 1994; Roston, 1994; Rosenman, 1997; Rasheed *et al.*, 1997).

Field systems are quite different from industrial manipulators. The diversity in topology and need for mobility as well as manipulation prohibits the direct application of the above techniques to field systems. New methods are required. Preliminary work has been conducted in the modular design of field robots (Farritor *et al.*, 1996; Rutman, 1995).

In this paper a hierarchical selection process is proposed to search for the best robot assembly. The proposed methodology 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 (Farritor *et al.*, 1996; Rutman, 1995).

(******) Another method uses a genetic algorithm to find configurations that allow the robot to reach designated targets while avoiding obstacles (Chocron and Bidaud, 1997).

3 The Modular Design Problem

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

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 Design versus Modular Design

In important ways, the design of a modular system can be simpler than the design of a conventional system. Conventional design variables are in general continuous, and the number of possible solutions is infinite. In modular design the design space is discrete. This places an upper bound on the size of the modular design space. Theoretically, this space could be enumerated and every possible design evaluated. As shown in Section 3.2, the number of possible solutions in this discrete space grows very rapidly with the number of available modules. For any real problem an exhaustive evaluation of these solutions is not practical. Also, since modular designs consist of pre-existing components their characteristics are known a priori.

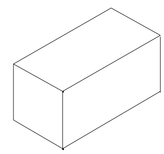
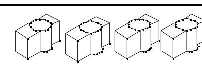
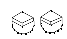
3.2 The Modular Robot Design Space

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:

1) All robot assemblies must contain a power/control module.
2) All modules are assembled in a serial chain called limbs.
3) All limbs are attached to ports on the power/control module.
4) All limbs must terminate in a module classified as an end effector.
5) All modules in the inventory do not need to be used in producing a design.

Consider the following simple example of systems constructed from the inventory shown Table 1. Here, 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 on the power module that limbs can be attached, called ports.

Table 1: A Simple Inventory

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

The number of possible designs, D, is the product of two factors: the number of limbs that can be created and where these limbs can be placed on the power module.

$$D = [(\text{moving limbs amongst the ports}) \leftrightarrow (\text{number of possible limbs})] = D_{ports} x D_{limbs} \qquad (1)$$

where:

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

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

Where D_{limbs}, is the number of possible limbs, 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 in this assembly is equal to the number of end effectors used. D_{ports} is the number of assemblies that can be created using i limbs. These different robots result from moving the i limbs to different ports on the power control module.

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).

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

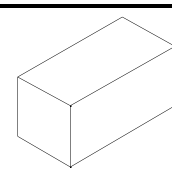
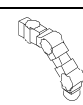
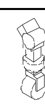
The size of that search space for various numbers of joints and end effectors is shown in Table 2. The search space for the small inventory of Table 1 contains 2800 possible robots. The modular design space, even for such a simple inventory, grows rapidly with the number available modules. For a more realistic inventory with 14 joints, 8 connecting links and 7 end-effectors can produce over 10^{20} robots (Farritor, 1998-1).

Table 2: Number of Robot Assemblies

	n_{feet} →	1	2	3	4	5	6
n_{joints}	4	70	2800	7.9×10^4	10^6	10^6	10^8
	6	98	5194	1.4×10^5	10^6	10^8	10^9
	8	104	8316	3.6×10^5	10^7	10^8	10^9

The inventory could consist of limbs, or *higher-level modules*, instead of individual modules, see Table 3. The design space for an inventory using higher level

Table 3: An Inventory with Higher-Level Modules

Number	Name	Quantity	
1	Electrical Power/Control Module	$n_1 = 1$ $N_{ports} = 14$	
2	Limb A	$n_2 = 6$	
3	Limb B	$n_3 = 6$	

modules is dramatically smaller. 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. The number of possible assemblies that can be created in using this inventory is given by (Farritor, 1998-1):

$$D = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \left(\frac{N_{pos}!}{i!\,j!\,\left(N_{pos} - i - j\right)!} \right) \qquad (5)$$

The number of designs that can be produced with the inventory of Table 3 is 3.34×10^7, compared to the 10^{20} designs from an inventory using low-level modules. 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}. This observation is utilized in the hierarchical design approach.

4 The Hierarchical Design Approach

A Hierarchical Selection process is proposed to search for the best assembly in the modular design workspace. The key to the proposed methodology is to apply physically based rules to reduce 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. 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; Rutman, 1995). 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 the later, more computationally intensive tests.

The process consists of tests and filters various levels. It eliminates entire sub-trees of solutions from further consideration, see Figure 2. The tests and filters exploit the physical nature of the system and the task.

At the first level, individual modules are considered. 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. Hence, filters at the early stages are very effective in reducing the size of the design space later in the process.

At a second level a group of modules, or sub-assembly, can be considered. For example, sub-assemblies that contain no joint modules can never produce a useful robots, see Figure 2. Therefore, this entire branch of solutions is eliminated.

4.1 The Assumptions of the Hierarchical Selection Process

To design a robot for a task there must be a method of determining if the robot is “good” or “bad” at performing the task. It is not practical to construct all candidate robots and evaluate their performance or even simulate them because of the large number of possible assemblies. Here, it is assumed that computationally simple tests can help distinguish between “good” and “bad” designs.

It is also assumed that a robot can be designed without precise knowledge of how it will execute the task. This assumption essentially de-couples the design and planning problems. However, the final stages of the design process may require some iteration between selecting a design and developing its plan.

With these assumptions it cannot be guaranteed that the method will produce an optimal design. This would require an exhaustive search. To formally "prove" these assumptions would be very difficult, if not impossible. Here, the validity of the assumptions has been considered within the context of example applications. It is shown that the hierarchical search process is practical for finding a sufficient design.

4.2 Task and Inventory Descriptions

To design a robot a description of the task is required. In this method tasks are described by a combination of task primitives that are relevant to the class of tasks being considered. The class of tasks considered in this paper is inspection robots for pipe 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 are shown in Figure 4. These primitives are used to create the tests and filters of the selection process. Table 4 shows a set of simple tests derived from the task primitives. Other constraints can be also added to the task description such as the maximum robot cost or weight.

Table 4: Example Simple Tests

Task Requirement	Example Simple Test
1) Max. Applied Force	$F_{endpoint}$
2) Smallest Passage	X, Y, Z size
3) Tallest Step	Limb length limb strength
4) Widest gap	Limb length limb strength
5) Max. Payload	$F_{endpoint}$ all limbs
6) Max. Traverse	Available energy
7) Max. Grade	limb strength coefficient of friction
8) Min. Turn	X, Y, Z size
9) Max. Reach	Maximum limb length
10) Scale	limb strength coefficient of friction
Time to complete task	velocity w.r.t max. traverse

The module inventory is characterized before the design process begins. Table 5 shows the inventory used here. It includes power/control, joint, link, and end effector modules. Robots constructed from this inventory follow the assembly rules of Section

3.2. All robots must contain either a Module #001 or Module #004 for on-board control computers. Modules #002 and #003 can be attached to the rear of #001 to provide additional energy. Module #004 is a tethered pneumatic power module. Power modules provide ports where other modules can be attached. The ports provide an energy connection of one of two types, electric or pneumatic. Electric modules are not compatible with pneumatic and vice-versa.

Rotary joint modules are available with various sizes, strengths and speeds. They can be attached in two configurations, corresponding to a 90-degree rotation of their axis. All limbs must terminate with an end effector module. Wheels, feet, and grippers are included. A wheel and a gripper can be used as a foot. The ability to make robots that walk and roll creates diversity in the designs. The inventory also contains connecting link modules used for dimensional changes. They can be connected to modules of either energy type.

Table 5: Module Inventory

ID#	Type	Quantity	Weight (oz.)	Dimension (in.)	Available Energy	Cost ($)	Notes
Power and Control Modules							
001	Electric	1	48	8x4x4	10 AA Alkaline 2750 mA-hr	3750	14 limb attach points computation and control
002	Electric	1	16	3x4x4	10-C Alkaline 7800 mA-hr	50	4 additional ports power only
003	Electric	1	16	3x4x4	10-C Alkaline 7800 mA-hr	50	4 additional ports power only
004	Pneumatic	1	60	16x8x8	Tethered	4000	16 ports computation and control
Joint Modules							
101	Small (E)	6	1.5	2.25x1.5x1	.593	70	42 oz-in stall
102	Medium (E)	6	3.3	2.25x1x1	.585	110	92 oz-in stall
103	Large (E)	6	2.8	2.5x1.3x1.8	.54	240	200 oz-in stall
105	Non-backdrive(E)	4	2.8	2.5x1.3x1.8	.54	240	300 oz-in stall
151	Small (P)	6	5.5	1x3x1	Tethered	125	200 oz.-in. stall
152	Medium (P)	6	6.2	1.5x4x2	Tethered	125	325 oz-in stall
153	Large (p)	6	8.0	2x6x3	Tethered	180	580 oz-in stall

End Effector Modules							
301	Rubber foot	8	.25	1x1x1	0	5	
302	Magnetic foot	8	.65	1x1x1	.75	35	16 oz. break-away force
303	Suction cup	8	.25	1x1x1	0	5	10 oz. break-away force
304	Wheel	6	3.5	2.5x2.5x1	.54	180	150 oz.-in. stall D_{wheel} = 2”
305	Track	4	15.0	2.5x2.5x4	1.2	250	150 oz.-in. stall
306	Gripper (E)	1	1.5	1.5x1x2	.8	210	.6 lbf grip
307	Gripper (P)	1	18.0	2x2x3	Tethered	150	6 lbf grip
Link Modules							
201	small	12	.5	1x1x1	NA	10	
202	medium	12	1.0	1.5x1x1	NA	10	
203	large	12	2.0	2x1x1	NA	10	

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

4.3 Module and Sub-Assembly Level Evaluations

The selection process begins by applying module filters derived from the task and inventory descriptions. They 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 6 shows some example the module-level filters and tests. External constraints such as maximum weight and cost are part of the task description. If a module exceeds these constraints it is eliminated. Geometric constraints, derived from the task primitives, are also used. Finally, all modules must have a power source of the correct energy type.

Table 6: Sample Module Filters and Tests

External Module Filters
- a) module weight < W_{max}
- b) module cost < C_{max}

Geometric Module Filters
- c) module size < l_{max}
- d) gripper span > d_{object}

Function Module Filters
- e) gripper force > W_{object}

Module Energy Domain Filters f) discard all modules without appropriate power sources g) discard all power sources without appropriate modules

Next, the design is analyzed on the sub-assembly level. Entire sub-assemblies and groups of sub-assemblies can be eliminated from consideration. For example, it might be decided that all sub-assemblies that do not contain joints will not produce useful robots. More complex tests can also be applied to groups of sub-assemblies such as sub-assemblies that contain three joints, see Figure 6.

Useful performance measures can be developed for sub-assemblies. For instance, the size of the limb workspace. A sub-assembly Jacobian can be developed. Information such as the maximum applied force, nominal power consumption per unit applied force, or maximum endpoint velocity can then be computed. The some sub-assembly tests used in this paper are shown in Table 7.

Table 7: Example Sub-Assembly Filters and Tests

Filters a) all sub-assemblies must end with an end effector b) cost < C_{max} c) weight < W_{max} d) maximum of 3 joints per limb (15 kinematic possibilities) Kinematic Analysis e) x reach f) y reach g) z reach h) DOF/dimension of workspace (1-D, planer, spatial) i) $F_{max} = [F_x \; ; \; F_y \; ; \; F_z]$ Power Analysis j) average power consumed Mobility Analysis k) Power / (Velocity*Weight)

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 demonstrated in Section **Error! Reference source not**

found.. By evaluating the designs at the sub-assembly level the search space is greatly reduced.

4.4 Assembly-Level Evaluation

Finally, the design process considers a complete robot assembly. First, simple filters such as cost, size, and weight are used. Examples of assembly evaluations can be seen in Figure 7 and in Table 8.

Table 8: Example Assembly Filters and Tests

Filters
a) cost < C_{max}
b) weight < W_{max}
Kinematic Analysis
a) static stability
a) x reach
b) y reach
c) z reach
d) Fmax = [F_x ; F_y ; F_z]
Power Analysis
e) average power consumed
f) peek power constraints
g) power to move
h) operating time
Mobility Analysis
i) DOF analysis
j) velocity analysis
k) Power / (Velocity*Weight)
l) max. distance that can be traveled

With the design space substantially reduced, but still large, a genetic algorithm (GA) is used to search for the best designs (Goldberg, 1989). The GA represents assemblies by a tree structure, or chromosome, see Figure 8. 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 (Goldberg, 1989).

The genetic algorithm evaluates robots using a fitness function. The tests and filters shown in Table 7 are used to produce a fitness value that estimates a robot's performance. Estimates are made for characteristics such as power consumption, applicable forces, static stability, and mobility. As an example, mobility is estimated using the average limb length with respect to the size of obstacles in the task. Sections 5 and 0 contain example fitness functions.

5 The USS Constitution Inspection Task

Here the modular design approach is used to develop a robot to assist in the conservation of the USS Constitution, a historic naval warship (Dubowsky *et al.*, 1995; Cole *et al.*, 1994; Roach). The robot is to inspect the area of the hull beneath the ship's ballast for dampness and rot-inducing conditions, Figure 9. It is impossible for workers to reach this area. The robot would enter through an access port and carry a video camera and mositure sensors. The structural members of the hull divide the sub-ballast area into a series of compartments. A typical compartment requires the robot to climb a 4.5" step and fit through the 8" access port. Table 9 shows the task primitives for this problem and their simple tests. Since each compartment has a different size ans shape, this is an excellent application for modular robots, as each compartment requires a somewhat different robot.

Table 9: Task Primitives for the USS Constitution Task

Task Requirement	Simple Test	Quantity
2) Smallest Passage	Y size Z size	8.5” 8.5”
3) Tallest Step	limb length limb strength	4.5” max(F_{z-max})
5) Max. Payload	Fendpoint all limbs	Max(F_{z-max})
6) Max. Traverse	energy margin	min(t_i)
8) Min. Turn	y size x size	11” 11”
Time to complete task	velocity w.r.t max traverse	d_{max}/Velocity

The robot should weigh less than 20 lb. to allow easy deployment and should cost less than $8000. It should be self-contained and operate for one-half hour.

The hierarchical selection process begins on the module-level. For example a test that requires all modules to fit through the access port eliminates the pneumatic power supply. This in-turn eliminates the pneumatic joints and the pneumatic gripper.

Sub-assembly evaluations were performed using the tests contained in Table 7 and Table 9. The maximum step requirement favored strong limbs. Also, the self-contained requirement emphasized the average power consumption of a sub-assembly.

The assembled robot designs are considered using the filters andtests from Table 8 and Table 9. Finally, the genetic algorithm searches 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 Equation (6). Where p_i is a number between 0 and 1 that estimates robot performance and w_i is a weighting factor used to assign relative importance.

$$f = \sum_{i=1}^{n} w_i p_i \qquad (6)$$

An example performance characteristic is the robot's climbing ability. It is estimated using the length of the robot's legs. For instance, a robot with 2" legs will not be capable of climbing the required 4.5" step. Figure 10 shows the assigned fitness as a function of leg length. The highest fitness is given to leg lengths between l_{min} and l_{max}. A short-legged robot will have no chance to clmib the step. A very long legged robots will have difficulty with ceiling interference. So l_{min} is determined by the maximum step height in a compartment and l_{max} is half the lowest ceiling height. This test is applied to each leg of the robot and the results are averaged to create a fitness that estimates the robot's kinematic climbing ability. This test does not place a bias on the total number of legs. The fitness falls off exponentially on both sides of the acceptable region. This to avoid discontinuity in the fitness function and enhance the convergence of the genetic algorithm (Wallace, 1994).

This search performed a crossover operation at a 60% rate and a mutation at a 2% rate. A population of 100 individuals was used and approximately 2000 generations were required for convergence. The search was completed in approximately 110 minutes using a Sparc Classic Sun workstation.

Four legged robots with power/control module #001 and symmetric legs proved to have high fitness, see Figure 11. Walking machines were favored by the selection process for two reasons. The first is the 4.5" step that descends into the inspection area. The second is the relatively short distance the robot travels.

Three high-scoring robot configurations are are shown in Figure 11a). Each has a somewhat different leg length due to the use of a different numbers of link modules, see Figure 11a) 13b) and 13c).

Robot I had the highest fitness (18.1) largely because its limb length of 8.75" was within the desired range of 8-9" (set using compartment geometry). Robot II had the second highest fitness of 17.4. Its limb length of 9.75 falls outside the desired leg length. Robot II was also penalized by added weight and cost. Robot III had a fitness of 17.11- the lowest fitness of the three configurations. It received a lower score largely due to the limb length and additional cost and weight.

At the last stage, this greatly reduced design space (of three robots) is evaluated using a computer simulation. The first objective was to determine if the high-scoring candidate designs could successfully perform the task. The second was to learn if the fitness scores correlated with simulation performance. The last objective was to study robot performance using different action plans.

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, a dynamics were not considered. It was developed so that any robot made from the module inventory could be easily evaluated. It is also used for the duct inspection task presented in Section 0.

Power consumption is one of the key performance factors considered by the simulation. It is assumed the actuators are the dominant power consuming elements and power requiremnet is proportional to motor torque (Dubowsky *et al.* 1995). To estimate the motor joint torques the foot reaction forces are found. When the robot is taking a step and has one leg lifted, calculating the three remaining foot reaction forces is a simple statically determined problem. When all four legs are in contact with the ground, the problem becomes statically indeterminate so compliance is introduced at each contact point, see Figure 12.

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. Then static equilibrium yields:

$$F_z = 0 : \; F_1 + F_2 + F_3 + F_4 - W = 0 \qquad \textbf{(7)}$$

$$Mx = 0 : -F_1 y_1 + F_2 y_2 - F_3 y_3 + F_4 y_4 = 0 \qquad \textbf{(8)}$$

$$M_y = 0 : -F_1 x_1 - F_2 x_2 + F_3 x_3 + F_4 x_4 = 0 \qquad \textbf{(9)}$$

$$F_n = k_n d_n \qquad (10)$$

Where d_n is the compression of spring n, W is the weight of the robot, and x_n, y_n and z_n are the foot position defined with respect to the center of mass of the robot. Since the robot is a rigid body a fourth equation relating d_1 to d_4 is written. For instance, if the robot is on a flat surface all its feet must lie in a plane, Equation (11).

$$A(x_4 - x_1) + B(y_4 - y_1) + C(z_4 - z_1) = 0 \qquad (11)$$

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 equation (12).

$$\begin{matrix} T_1 \\ T_2 \\ T_3 \end{matrix} = \left[J(\theta_1, \theta_2, \theta_3)\right]^T \begin{matrix} F_x \\ F_y \\ F_z \end{matrix} \qquad (12)$$

Where $[T_1,T_2,T_3]$ is a vector of the joint torques, J is the Jacobian of the limb, and $[F_x,F_y,F_z]$ is a vector of the reaction forces at the foot.

The joint torques are then used to estimate power consumption. This information is also used to ensure that the required actuator torques do not exceed their limits.

5.1 System Performance

Robot performance is dependent on the manner that the task is executed (i.e. the action plan). The three robot designs were compared in two ways. In the first comparison, each robot performed the task in the same manner, using a generic action plan (Farritor, 1998-1). In the second comparison each robot used a plan specifically developed for that robot and task. A model-based planning method was used (Farritor, *et al.*, 1998-2).

The results for the three case study robots, each using the generic action plan, are shown in Table 10. These results suggest that the modular design process has not selected the best configuration for the task. Robot I, the highest scoring robot, cannot perform the task using this action plan. This is because its legs are too short to step into the inspection area using the generic plan. However, robot II, which scores lower than robot I in the search, performs the task. Robot III also completes the task, although it consumes almost 150% more power then Robot II.

The results for the robots executing the task with an action plan specifically developed for the robot and task are also shown in Table 10. These results emphasize that the performance of a given robot depends on the action plan used. This is clearly seen in the case of Robot I. It was unsuccessful using the generic action plan and successful using the specific plan. This exemplifies the trade-off between optimality and robustness. While the highest scoring ("best") robot gave the very best performance using a plan developed specifically for it, it was not robust with respect to the plan. In the cases of Robots II and III the power consumption was slightly reduced using the specific action plans.

Table 10: Simulation Results

	Search Fitness	Plan Used	Result	Power Consumed
Robot I	18.1	Generic Plan	Unsuccessful	-----
		Specific Plan	Successful	647
Robot II	17.4	Generic Plan	Successful	744
		Specific Plan	Successful	730
Robot III	17.1	Generic Plan	Successful	1119
		Specific Plan	Successful	1037

These results suggest that while the plan used by a robot will effect its relative performance, solving the design and planning problems in a de-coupled manner can produces good designs. Solving them in a coupled manner is not practical for these systems. However, this work does suggest that a de-coupled approach can be viable in some cases. This areas is really an open research question.

6 A duct work Inspection Task

Here, the modular design process is demonstrated on a duct work inspection task. Such duct networks are common in many industrial and municipal areas. Often, difficult areas to access need to be inspected. Candidate designs for a duct work inspection task are presented and tested using detailed simulation similar to that explained in Section 5. Then, a final design is chosen.

6.1 Problem Definition

An inspection of the surface of the duct is needed requiring the robot to travel throughout the duct network, see Figure 13. The robot will be inserted in the network at the left of the figure. It will need to travel down a 30° slope and make 90° turns in an 11" 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 Figure 4 and is summarized in Table 11.

Table 11: Duct Task Test Parameters

Task Requirement	Simple Test	Quantity
2) Smallest Passage	Y size Z size	11" 11"
3) Tallest Step	limb length limb strength	2" $\max(F_{z\text{-}max})$
4) Widest gap	limb length limb strength	8" $\max(F_{z\text{-}max})$
6) Max. Traverse	energy available	$\min(t_i)$
7) Max. Grade	limb strength coefficient of friction	$\max(F_{z\text{-}max})$ $\mu > \tan(30)$
8) Min. Turn	y size x size	11" 11"
9) Max. Reach	maximum limb length	8"
Time to complete task	velocity w.r.t. max traverse	d_{max}/Velocity

Again, the hierarchical selection process first looks at the design on the module level. 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 also eliminated.

Next, the design problem was considered on the sub-assembly level. 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 and the genetic algorithm searched the reduced design space. The candidate designs developed by the genetic algorithm are shown in

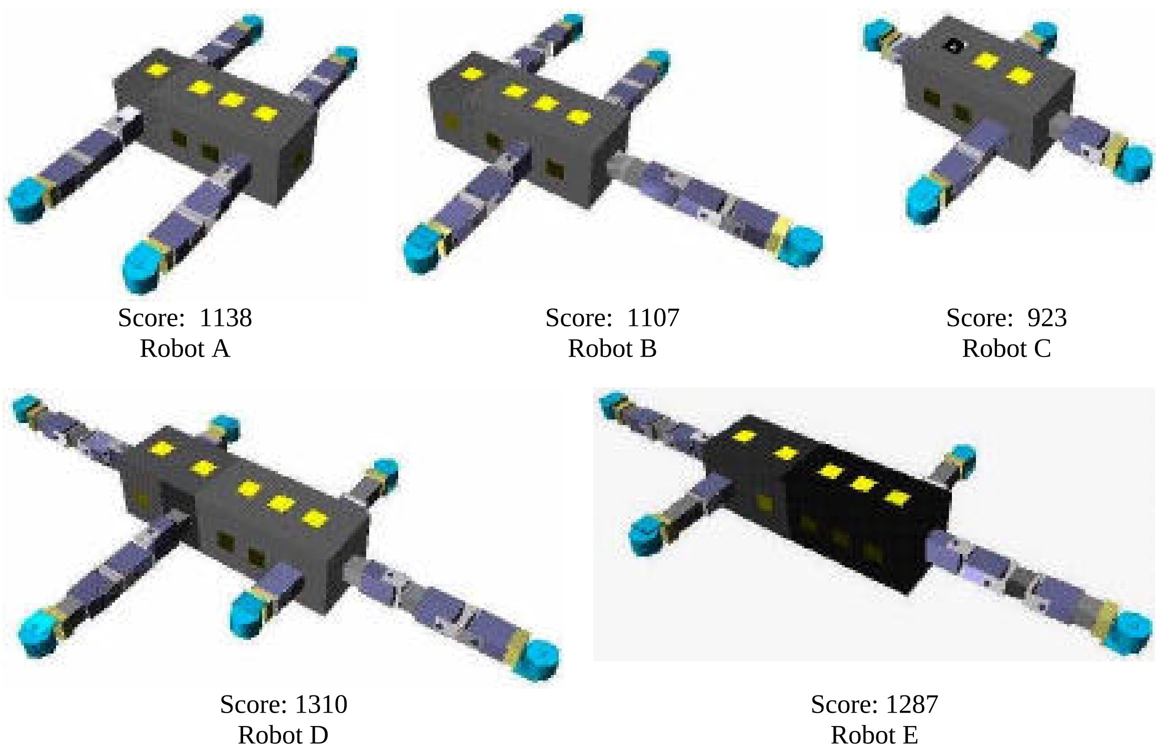


Figure 14. The selection process favored robots with wheels. The long distances to be traveled, along with the relatively simple climbing requirements, make wheels practical. Also, the requirement to cross an 8” gap caused the selection process to create long robots.

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. 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. A relative comparison of the designs is shown in Table 12 with Robot A used as the baseline.

Table 12: Relative Robot Performance

Parameter	A	B	C	D	E
Assembly weight	0	0	+	-	-
Assembly cost	0	0	+	-	-
leg length (average / stand. dev.)	0	-	-	-	-
x span	0	-	-	+	+
y span	0	+	-	-	-
Stability	0	-	-	+	+
Payload	0	0	+	+	+
Operating time	0	0	+	+	+
Maximum distance	0	0	+	+	+

Because of the similarity between Robots A and B and between D and E, Robots A and E were chosen for further evaluation. The simulation described in Section 5 was used. The robot was required to travel down the slope, turn right and cross the 8" gap. Then the robot must climb the 2" step into the narrow duct at the right of

Figure 13.

The robots were tested using an action plan developed specifically for the robot and task. The most challenging portion of the task 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 8" while maintaining static stability, see Figure 15.

However, the asymmetry and long span of Robot E allowed it to cross the gap as seen in Figure 16. In Frame 2, the first leg is extended across the gap. With this leg across the gap, the second leg can be lifted and the robot can advance (Frame 3). Now the second leg is planted and the third leg lifted (Frame 4). Finally, the forth leg can be lifted and the robot advanced (Frame 5). These results show that the asymmetry of Robot E is a good design for the duct work inspection task, while the more obvious solution (Robot A) was not successful.

7 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 example case studies 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. Primarily, the advantage of designing on the sub-assembly level was illustrated.

The design process was applied to two representative tasks. These tasks were used to explore the assumptions of the approach. First, detailed simulation results showed that useful robots could be constructed for different tasks from a moderately sized inventory of components. Also, simple tests were used to distinguish between "good" and "bad" robot designs. Successful solutions were found using the simple tests of the hierarchical selection process. Finally, the design and planning processes were de-coupled. Without consideration for the detailed plan that would allow the robots to accomplish the task, candidate robot designs were developed. Then, simulation showed the final robot designs were successful. The results of the case study support these assumptions.

References

Ambrose, R., Tesar, D., “Modular Robot Connection Design”, ASME 4th International Conference on Design Theory and Methodology, Scottsdale, AZ, 1992.

Benhabib, B. Zak, G. and Lipton, M., “A Generalized Kinematic Modeling Method for Modular Robots,” Journal of Robotic Systems, 1989. vol. 6, no. 5, pp. 545-71.

Chedmail, P., Ramstein, E., "Robot Mechanism Synthesis and Genetic Algorithms" in Proceedings of 1996 IEEE International Conference on Robotics and Automation, April 22-28, Minneapolis, MN, pp.487-94, 1996.

Chen I., Burdick J., "Determining task optimal robot assembly configurations", IEEE international conference on robotics and automation, pp. 132-37, 1995

Chen, I., Yang, G., “Automatic Generation of Dynamics for Modular Robots with Hybrid Geometry”, IEEE International Conference on Robotics and Automation, pp. 2288-93, 1997.

Chocron, O., Bidaud, P., “Genetic Design of 3D Modular Manipulators”, Proceedings of the IEEE International Conference on Robotics and Automation, pp. 223-28, 1997.

Cohen, R. Lipton, M., Dai, M. and Benhabib, B. “Conceptual Design of a Modular Robot.” Journal of Mechanical Design, Mar 1992. vol. 114, pp. 117-25.

Cole, J., Dubowsky, S., Rutman, N., and Sunada, C., “The Application of Advanced Robotics and Sensors to the Preservation of the USS Constitution” Conference on the Technical Aspects of Maintaining, Repairing and Preserving Historically Significant Ships, 1994.

Cole, J. Rapid Generation of Motion Plans for Modular Robotic Systems, Masters Thesis, Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 1995.

Dubowsky, S., *et al.*, “On the Design and Task Planning of Power Efficient Field Robotic Systems”, Proc. 6th ANS Robotic and Remote Systems, Monterey, CA, 1995.

Dubowsky, S., Cole, J., Rutman, N., and Sunada, C., “Mobile Autonomous Robotic Systems for Unstructured Environments with Applications to the USS Constitution” Proceedings of the NSF Design and Manufacturing Grantees Conference, San Diego, CA, 1995. Invited.

Farritor, S., On Modular Design and Planning for Field Robotic Systems, Ph.D. Thesis, Department of Mechanical Engineering, MIT, Cambridge, MA, May, 1998.

Farritor, S., Hacot, H., Dubowsky, S., "Physics-based Planning for Planetary Exploration," IEEE Conference on Robotics and Automation, May 1998.

Farritor, S., Dubowsky, S., Rutman, N., "On the Rapid Design of Field Robotic Systems", ASME Conference on Design Theory and Methodology, August 1996.

Goldberg, D. Genetic Algorithms in Search, Optimization, and Machine Learning. Addision-Wesley, Reading, MA 1989.

Hayati, S., et. al., “The Rocky 7 Rover: A Mars Sciencecraft Prototype”, Proceedings of the 1997 IEEE International Conference on Robotics and Automation, pp. 2458-64, 1997.

Hirose, S. Biologically Inspired Robots: Snake like Locomotion and Manipulation. Oxford Scientific Publications, 1993.

Kelmar, L., and Khosla, P. “Automatic Generation of Forward and Inverse Kinematics for a Reconfigurable Modular Manipulator System,” Journal of Robotic Systems, 1990. vol. 7, no. 4, pp. 599-619.

Kim, J., and Khosla, P. "Design of Space Shuttle Tile Servicing Robot: an Application of Tasked Based Kinematic Design," Proceedings of the IEEE International Conference on Robotic and Automation, Atlanta, GA, May 1993. vol. 3, pp. 867-74.

Matthies, L., Gat, E., Harrison, R., Wilcox, B., Volpe, R., Litwin, T., "Mars microrover navigation: performance evaluation and enhancement," Autonomous Robots, 2(4), 1995.

Nagakubo, A., Hirose, S., "Walking and Running of the Quadruped Wall-Climbing Robot," IEEE International Conference on Robotics and Automation, pp.1005-12, 1996.

Pamecha, A.,Chiang, C., Stein, D., Chirikjian, G.S., “Design and Implementation of Metamorphic Robots”, Proceedings of the 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference, Irvine, California, August 1996.

Paredis, C. J. J., Brown, H. B., Khosla P. K., "A Rapidly Deployable Manipulator System" IEEE International Conference on Robotics and Automation, April 22-28, pp. 1434-39, 1996.

Paredis, C., and Khosla, P., "Synthesis Methodology for Task Based Reconfiguration of Modular Manipulator Systems” Proceedings of the 6th International Symposium on Robotics Research (ISRR ‘93), Hidden Valley, PA, October 2-5, 1993

Paredis, C., An Agent-Based Approach to the Design of Rapidly Deployable Fault Tolerant Manipulators, Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, August 1996.

Pratt, J., Dilworth, P., Pratt, G., “Virtual Model Control of a Bipedal Walking Robot”, IEEE International Conference on Robotics and Automation, Albuquerque, NM, 1997.

Rasheed, K., Hirsh, H., Gelsey, A., “A Genetic Algorithm for Continuous Design Search Spaces”, Artificial Intelligence in Engineering, pp. 295-305, 1997.

Roach, John Charles. *Old Ironsides: an Essay in Sketches* Department of the U. S. Navy

Rosenman, M., “An Exploration into Evolutionary Models for Non-Routine Design”, Artificial Intelligence in Engineering, pp. 287-93, 1997.

Roston, G., A Genetic Methodology for Configuration Design, Carnegie Mellon University Technical Report, CMU-RI-TR-94-42

Rutman, N., Automated design of Modular Field Robots, Masters Thesis, Department of Mechanical Engineering, MIT, Cambridge, MA, 1995.

Schenker, P., et. al., “Lightweight Rovers for Mars Science Exploration and Sample Return,” Intelligent Robots and Computer Vision XVI, SPIE Proc. 3208, Pittsburg, PA, October, 1997.

Shiller, Z., and Dubowsky, S. "On Computing the Global Time Optimal Motions of Robotic Manipulators in the Presence of Obstacles," IEEE Journal of Robotics and Automation, Dec. 1991. vol. 7, no. 6, pp. 785-97.

Tesar, D., and Butler, M. "A Generalized Modular Architecture for Robot Structures," Manufacturing Review, 1989. vol. 2, no 2.

Wallace, D., A Probabilistic Specification-based Design Model: applications to design search and environmental computer aided design, Ph.D. Thesis, Department of Mechanical Engineering, MIT, Cambridge, MA, 1994

Weisman, R., “GRI’s Internal Inspection System for Piping Networks,” Proc., 40th Conference on Remote Systems Technology, Vol. 2, 1992, pp. 109-15.

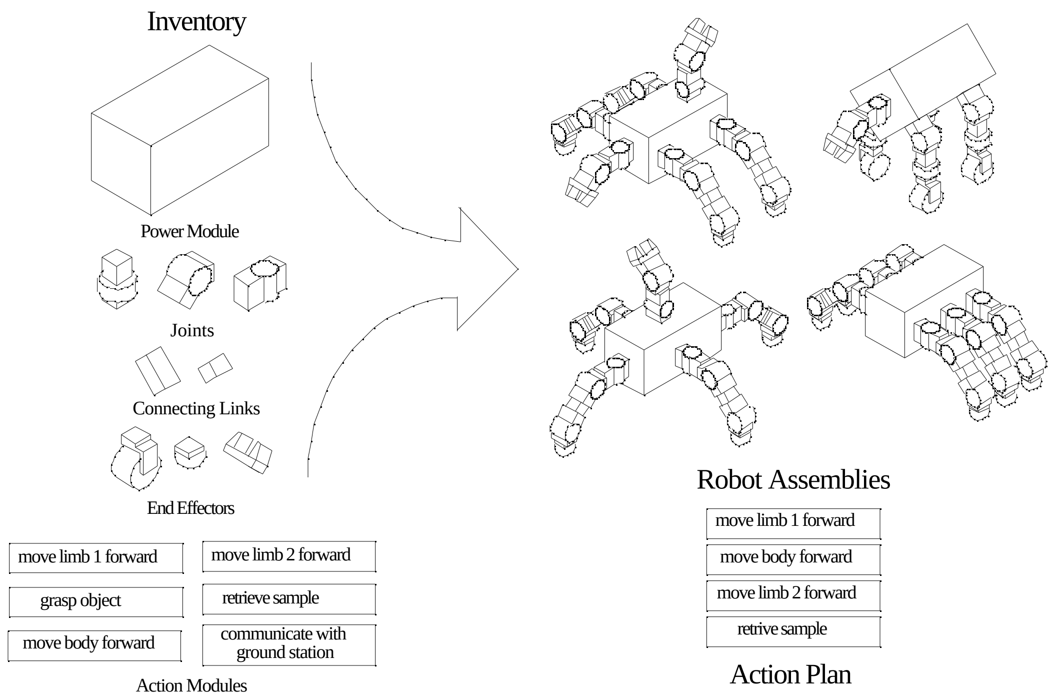


Figure 1: A Modular Approach to Field Robotics

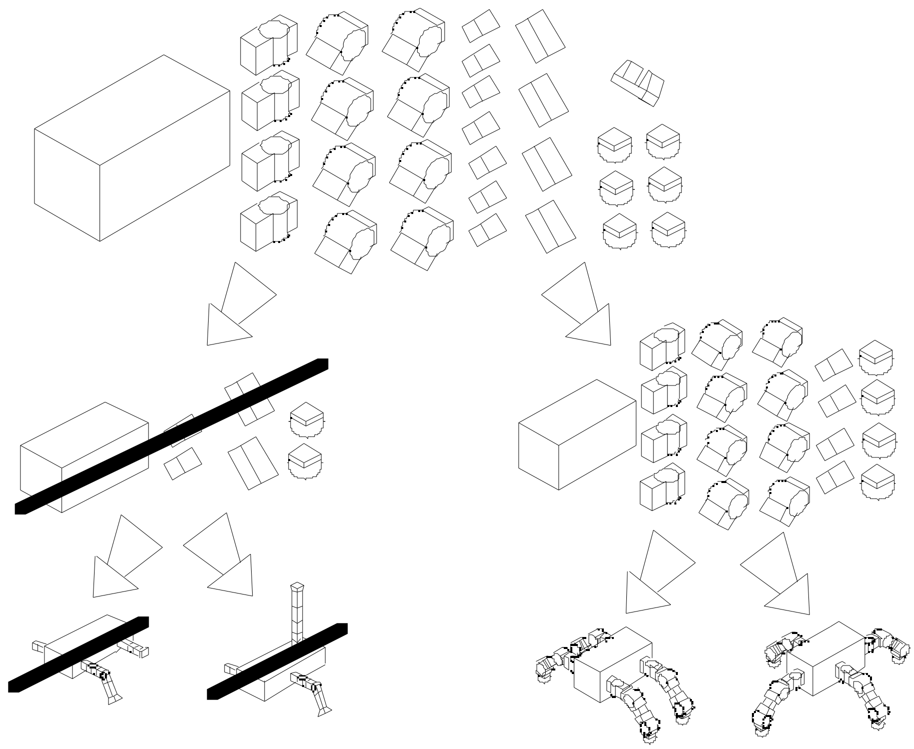


Figure 2: Hierarchical Selection Process

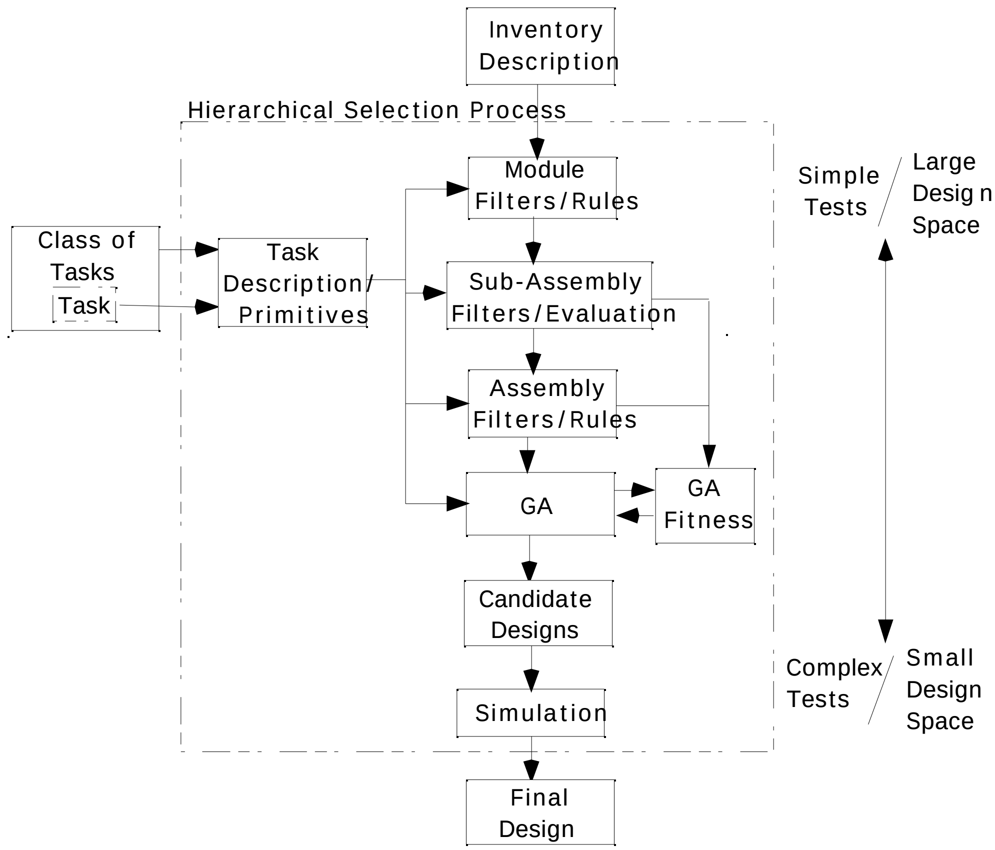


Figure 3: The Modular Robot Design Structure

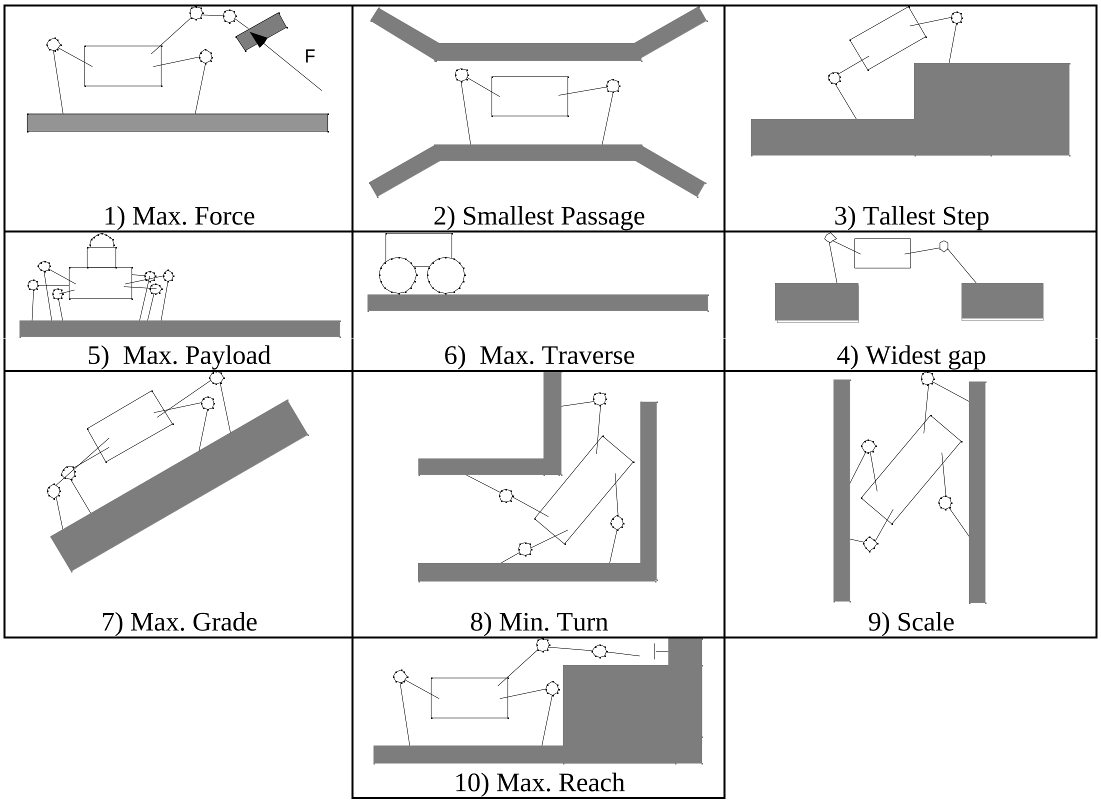
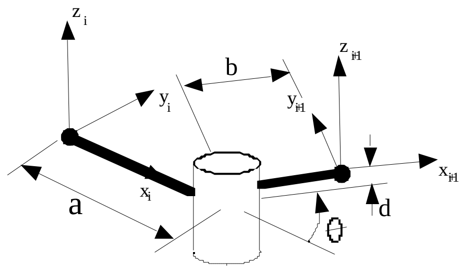


Figure 4: Task Primitive Inventory

JOINT KINEMATIC MODEL:



$$T_i^{i+1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & a + b\sin(\theta) \\ \sin(\theta) & \cos(\theta) & 0 & b\sin(\theta) \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

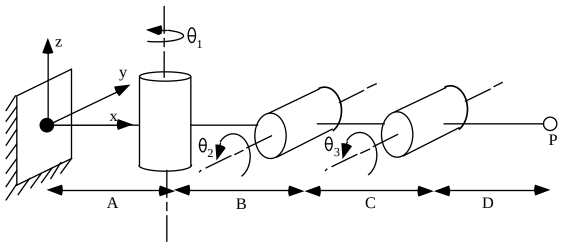
a = 1.50”
b = 1.00”
d = 0.00”

JOINT MODULE SPECIFICATIONS:

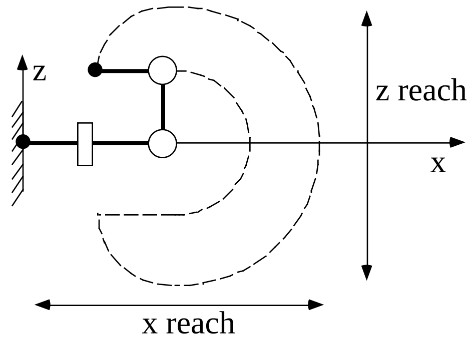
Stall Torque = 42 [oz.-in.]
Weight = 1.5 [oz.]
Cost = $70
Gear Ratio = 56:1
Motor Inductance = 126 [mh]
Torque Constant = 536 [oz.-in./amp]
Range of motion = $\{-95° < \theta < 95°\}$

Figure 5: Joint Module Description

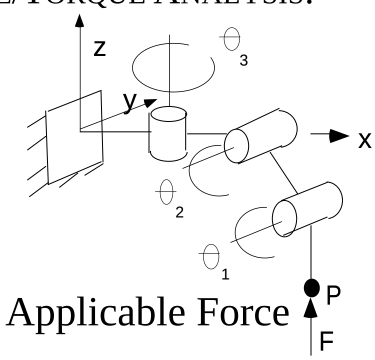
KINEMATIC CONFIGURATION:
3 Joints (Z-Y-Y)



WORKSPACE:



FORCE/TORQUE ANALYSIS:



VELOCITY ANALYSIS:

$$J = \begin{matrix} -Bs_1 - Cc_2s_1 - Dc_{23}s_1 & -Cs_2c_1 - Ds_{23}c_1 & -Ds_{23}c_1 \\ -Bc_1 - Cc_2c_1 - Dc_{23}c_1 & -Cs_2s_1 - Ds_{23}s_1 & -Ds_{23}s_1 \\ 0 & Cc_2 - Dc_{23} & Dc_{23} \end{matrix}$$

Figure 6: Kinematic Sub-Assembly

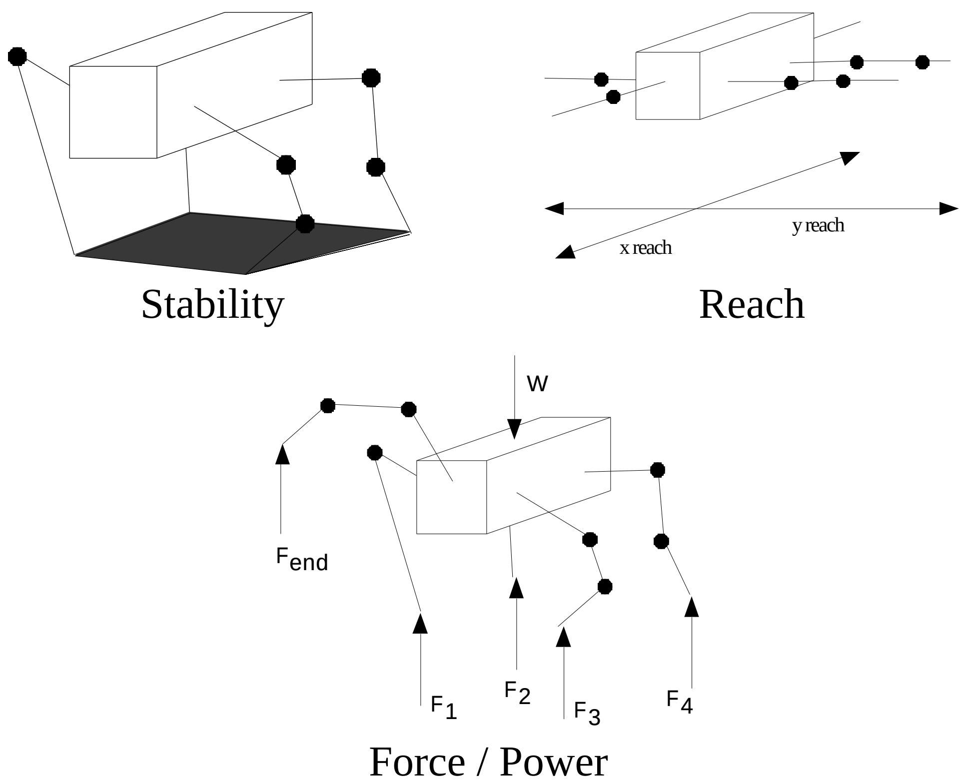
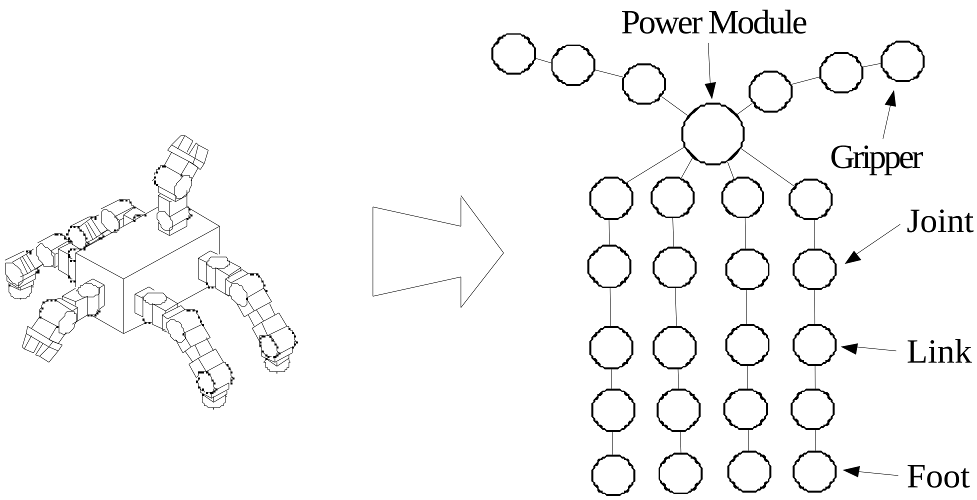
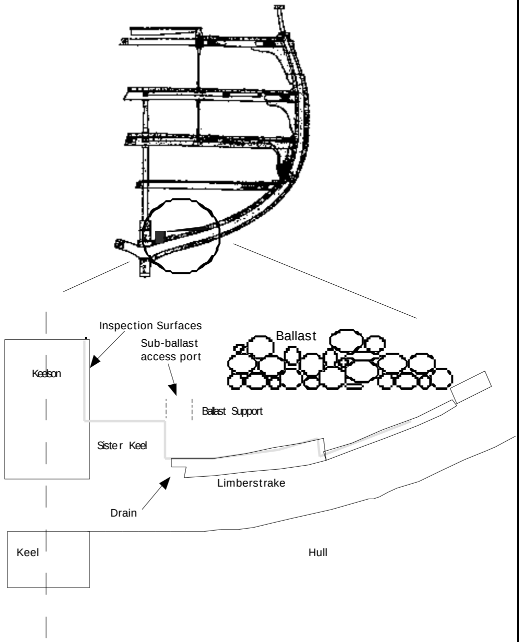


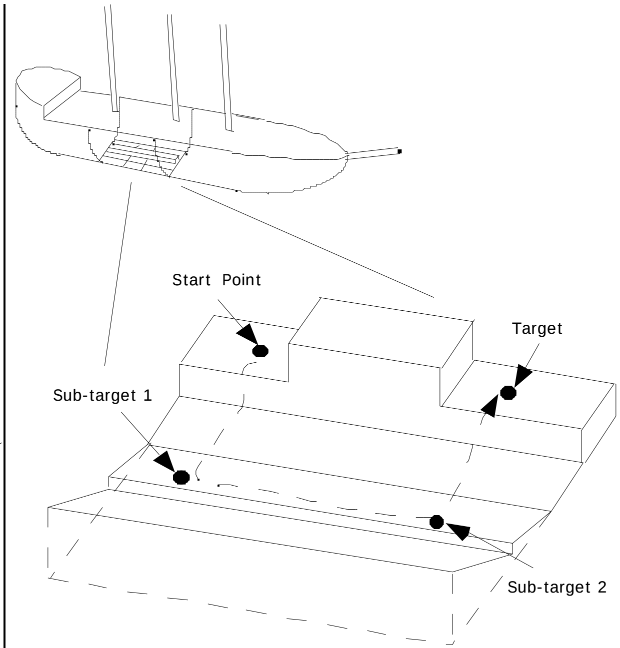
Figure 7: Assembly-Level Evaluation



Robot Assembly Tree Chromosome Representation

Figure 8: Genetic Representation





a) Inspection Area Cross-Section (Rutman, 1995)

b) Model of Inspection Area (without ceiling)

Figure 9: USS Constitution Inspection Task

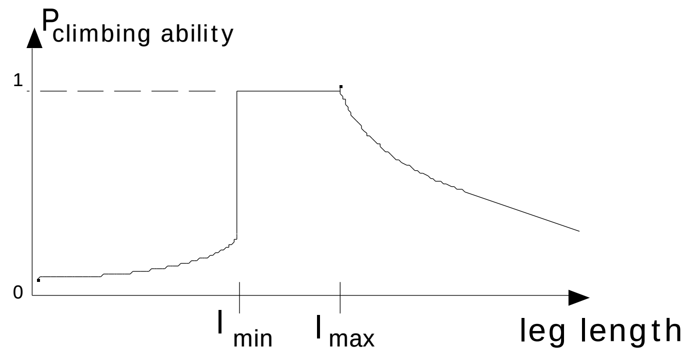
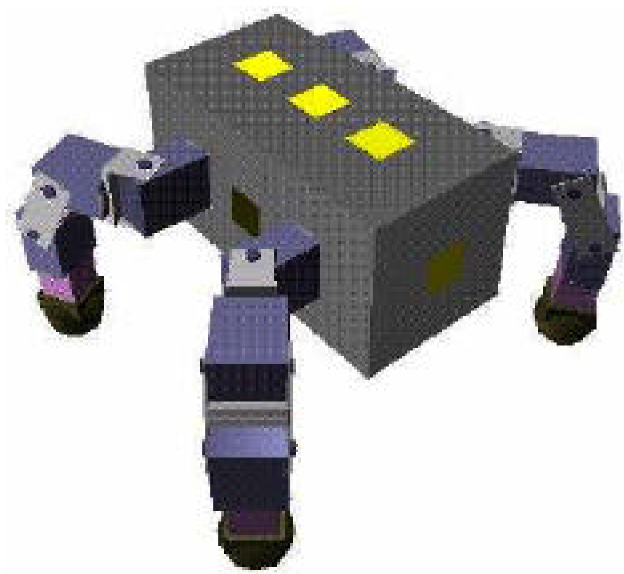
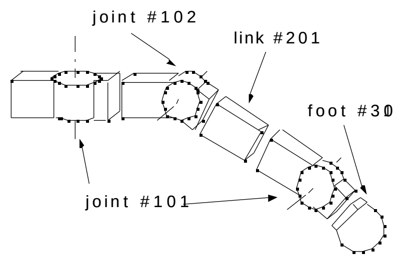
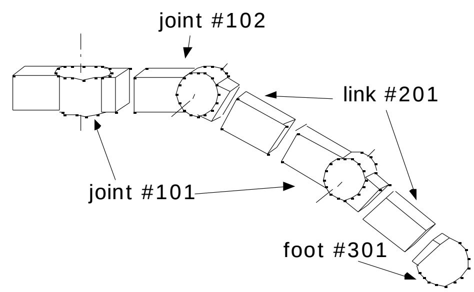


Figure 10: Limb Length Fitness

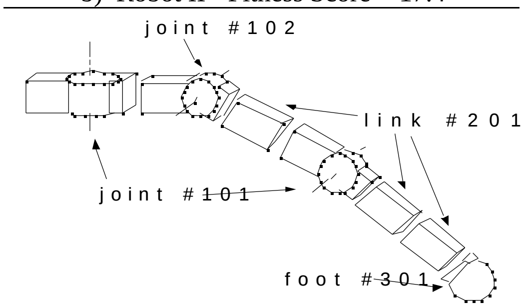




a) Robot I - Fitness Score = 18.1



b) Robot II - Fitness Score = 17.4



c) Robot III - Fitness Score = 17.11

Figure 11: Candidate Design's Limb Configurations

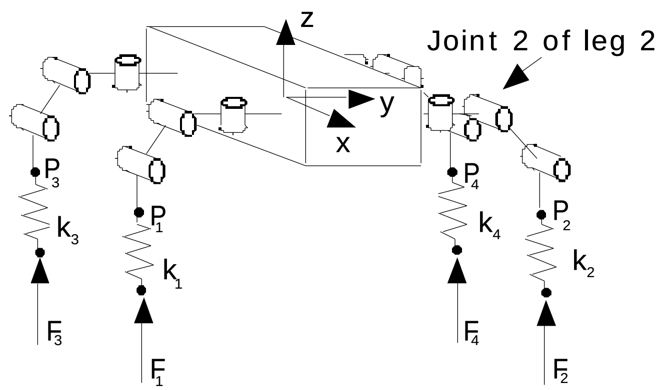


Figure 12: Calculation of Reaction Forces

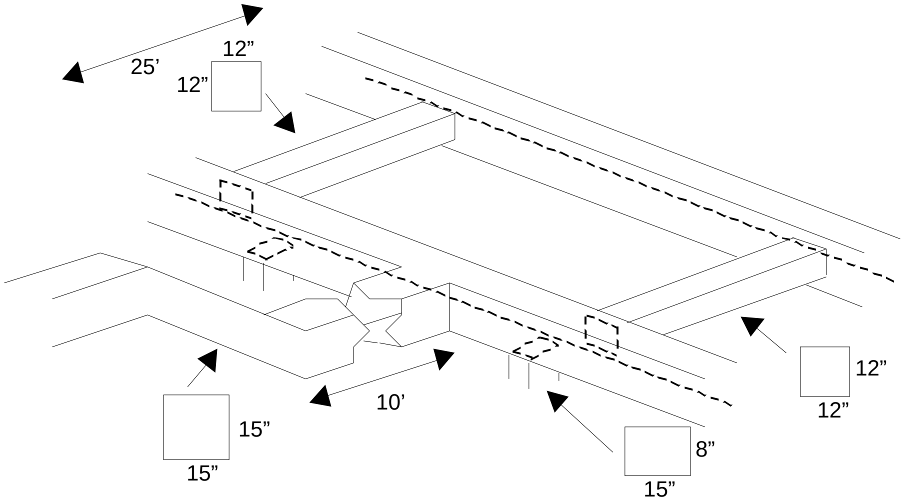
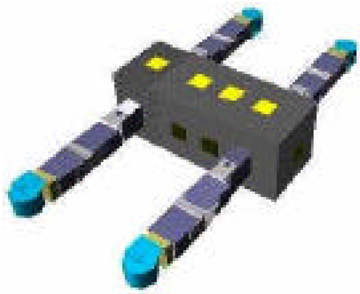
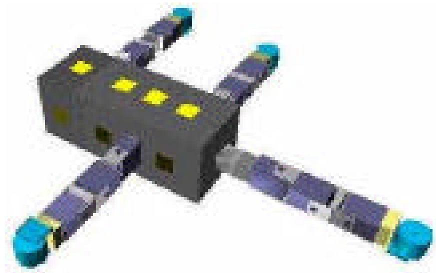


Figure 13: Duct Work Inspection Task

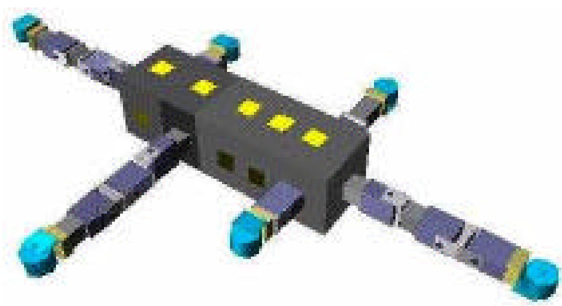


Score: 1138
Robot A

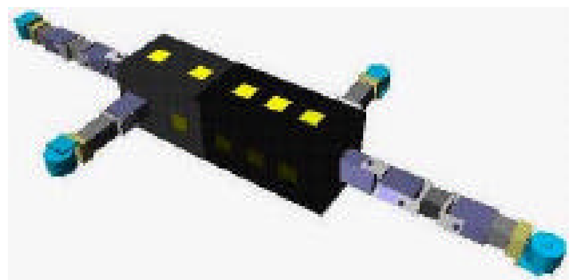


Score: 1107
Robot B

Score: 923
Robot C



Score: 1310
Robot D



Score: 1287
Robot E

Figure 14: Candidate Robot Designs

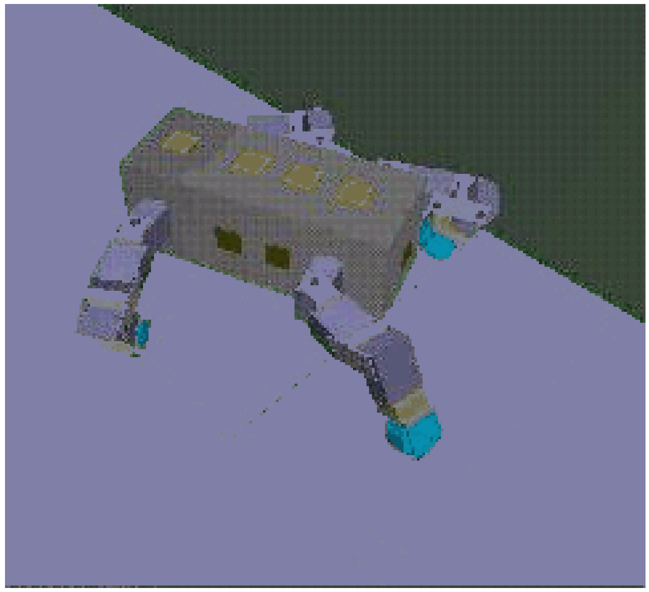
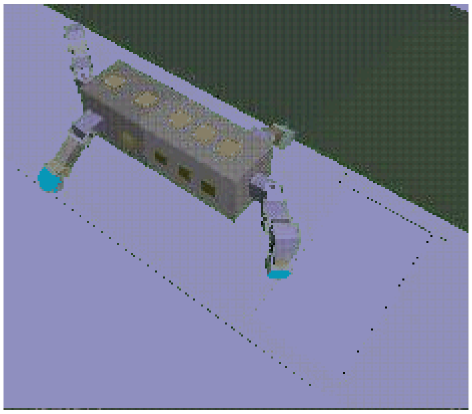
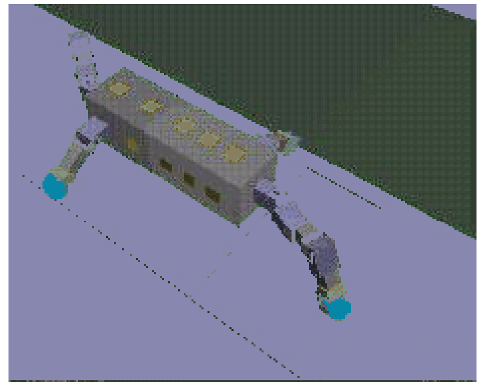


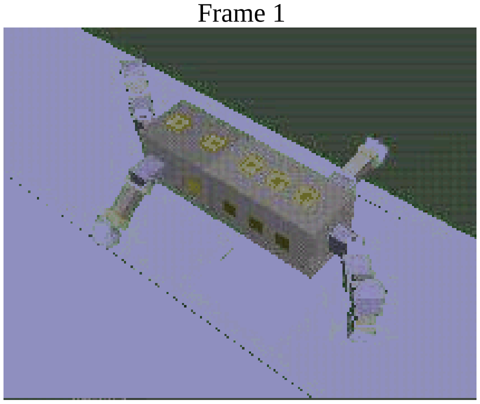
Figure 15: Robot A - Crossing the Gap



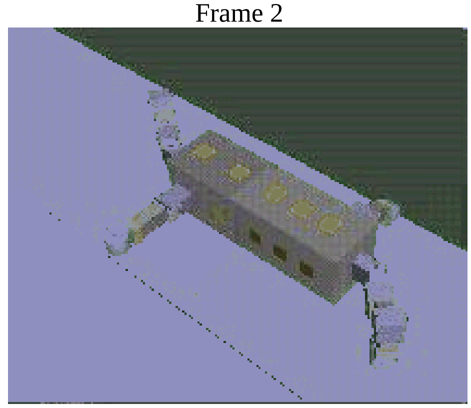
Frame 1



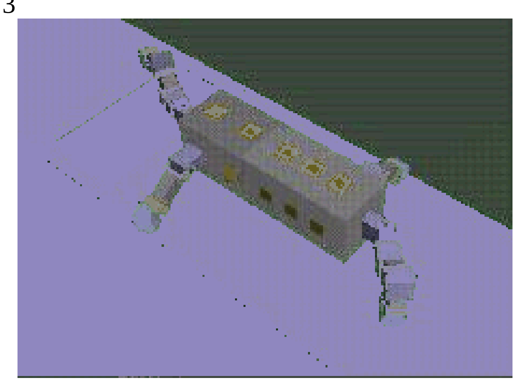
Frame 2



Frame 3



Frame 4



Frame 5

Figure 16: Robot E - Crossing the Gap