Robot modularity for self-reconfiguration

Peter Will Andrés Castaño Wei-Min Shen

Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292

ABSTRACT

Metamorphic robots are an emerging field in which robots can dynamically reconfigure shape and size not only for individual robots (intra-robot metamorphing) but also for complex structures that are formed by multiple robots (inter-robot metamorphing). Such capability is highly desirable in tasks such as fire fighting, earthquake rescue, and battlefield scouting, where robots must go through unexpected situations and obstacles and perform tasks that are difficult for fixed-shape robots. This research direction presents a number of technical research challenges. Specifically, metamorphic robots must be able to decompose and reassemble at will from a set of basic connectable modules. Such modules must be small, self-sufficient and relatively homogeneous. In this paper, we present our approach to address these issues and describe the design of the CONRO modules. These modules are equipped with a low power micro-processor, memory chips, sensors, actuators, power supplies, and miniature mechanical connectors used for communication and power sharing. We will also describe a set of control mechanisms (including one that is inspired by the biological concept of "hormone") for controlling gaits and reconfigurations. We conclude the paper with a status report of the CONRO project and a list of the future work needed to fully realize the construction of the CONRO metamorphic robots.

Keywords: Reconfigurable robot, miniature, autonomous, robot module

1. INTRODUCTION

The CONROs (CONfigurable RObots) are metamorphic robots (i.e., shape-changing robot) designed for real-world applications such as fire fighting, urban search and rescue after an earthquake, and battlefield reconnaissance. These robots react to challenging tasks by changing their shape and size. They can (de)compose themselves into task-directed robots to address unexpected situations and obstacles or perform tasks that are difficult for fixed-shape robots. For example, CONRO robots such as those shown in Fig. 1, may become snakes to maneuver through difficult terrain, to travel down a pipe or to cross a wired fence, or they may grow legs to turn into hexapods able to climb stairs and travel on uneven terrain. These robots are built using sets of identical robotic modules. Each module is a self-sufficient miniature robot with sensors, actuators, micro-processor, battery and communication capabilities. As shown in Fig. 2, the extremes of each module has sets of connectors (one on each shaded face), that allows it to connect or disconnect to other modules at will. Because the modules are identical and independent from each other, they can connect not only to other modules of the robot (intra-robot metamorphing) but also to modules of other robots, creating in the process larger robots (inter-robot metamorphing). For example, a CONRO robot may disassemble itself into a set of small robots that crawl under the door of a closed room. Once inside, they may reassemble into the original robot or assemble a set of wheels and build the necessary locomotive and sensory components needed for a given task, such as carrying a heavy payload.

Metamorphic robots offer a new vision where large scale results may be accomplished by the coordinated actions of a large numbers of small robots. These robots can self-assemble and/or reconfigure into new body shapes with locomotive and sensory primitives suitable for different tasks. CONRO metamorphing can be performed at two different levels. At the intra-robot level, a single metamorphic robot must be able to change its shape and size by rearranging its body parts. At the inter-robot level, a group of metamorphic robots should be able to join and coalesce into a larger and more complex robot, or a single large robot should be able to decompose into a set of smaller but more agile robots. These tasks present a number of technical challenges for the current robotics research. Specifically,

Further author information: (Send correspondence to Peter Will)
Peter Will: E-mail: will@isi.edu

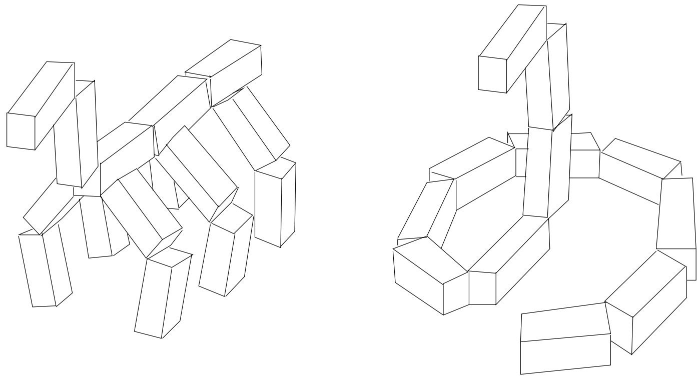

Figure 1. A hexapod and a snake CONRO robots.

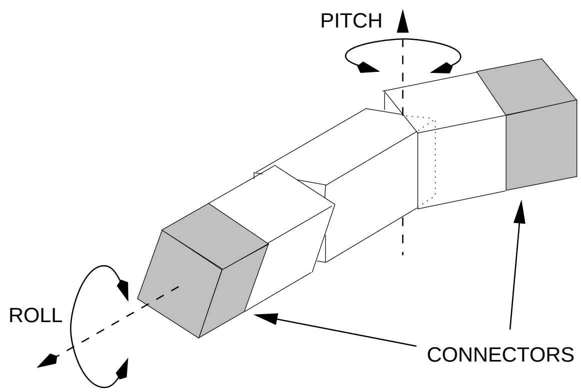

Figure 2. Module of CONRO.

a metamorphic robot must be able to decompose and assemble at will from a set of basic connectable modules. These modules must be small (the actuator in a module must be able to lift several other modules), self-sufficient (each module must be able to see, act, and think during reconfiguration and task mission), and relatively homogeneous (no single damaged module should paralyze the entire system). Furthermore, the modules of the metamorphic robots must maintain communication and collaborate in the execution of complex tasks (e.g., locomotion, reconfiguration, analysis of sensor information), all within the intrinsic limitations of any single module. All these issues (i.e., miniaturization, distributed control, autonomy) are technical challenges for today's robotic systems.

Metamorphic robots were first proposed by a number of robotics researchers. Fukuda and Kawauchi[1] proposed a cellular robotic system to coordinate a set of specialized modules. Yim[2] studied how to achieve multiple modes of locomotion using robots composed by a few basic modules. Murata et al.[3] and Yoshida et al.,[4] separately, designed and constructed systems that can achieve planar motion by arranging modules. Pamecha et al.[5] described metamorphic robots that can aggregate as stationary 2-D structures with varying geometry and that implement planar locomotion. Kotay et al.[6] proposed and implemented metamorphic robots based on a robotic module called the "robotic molecule". Nilsson[7] designed and implemented a torsion-free joint for modular snake-like robots. Fujita et al.[8] build a biologically inspired reconfigurable robot. Paredis and Khosla[9] proposed modular components for building fault tolerant multipurpose robots. Neville and Sanderson[10] proposed a module for the dynamic construction of complex structures.

The CONRO metamorphic robots present a number of unique features when compared to most existing work in reconfigurable robots. First, the CONRO modules are being designed to be miniature and self-sufficient. The miniaturization of the module gives its actuator a higher torque to weight ratio allowing it to lift (and therefore manipulate) a large number of identical modules. The self-sufficiency of the module is a requirement to have an autonomous robot. Second, the CONRO modules are homogeneous (as much as possible) to add redundancy to the robot (i.e., a fault tolerant mechanism that allows any module to be replaced by any other) and to ease the the reconfiguration process (reconfiguration can be accomplished with few connects/disconnects because every module has the same functionality). Third, in contrast to other reconfigurable robots where gait controls are mixed with reconfiguration (i.e., most reconfigurable robots must rearrange their body parts to move), the CONRO robots are able to separate locomotion from reconfiguration, and can select the best configuration for a given situation and move with configuration-dependent efficient gaits.

This paper is organized as follows. Section 2 lists and analyzes the capabilities required for a CONRO robot to be able to metamorph. Section 3 describes our approach to design and implement these capabilities. Section 4 reports the current status of the CONRO modules and presents some experimental results. Section 5 discusses the near and medium term future work. Finally, we present our conclusions in Sect. 6.

2. REQUIRED CAPABILITIES FOR METAMORPHIC ROBOTS

If we assume that metamorphic robots are made of modules that serve as their basic building blocks, then we must have a clear understanding of the capabilities that these modules must have. At a high level, a self-sufficient metamorphic robot must perform two distinct but closely related tasks: self-reconfiguration and "regular" robot tasks such as navigation, locomotion, and manipulation. These high level tasks are supported, at a low level, by hardware and software components. The hardware support is provided by the module connector devices, the actuators and sensors, the communication devices, the on-board micro-processor and the power supply. The software support is provided by programs that control the docking, motion (needed for both locomotion or manipulation), morphing, and by the monitor programs that provide fault tolerance. In this section, we discuss these capabilities and the technical challenges involved in their implementation.

One of the most difficult module components to design is the connector which must be reliable (connect and disconnect only when commanded to), power-efficient (use minimal power when necessary), and must provide an interface for power sharing and communication between modules. To the best of our knowledge, CONRO is the first attempt to build such a (dis)connector at a miniature scale. In the CONRO project, we assume that each module has a number of connectors at its extremes. Two connectors per module (i.e., one at each extreme of the module) allow the construction of linear structures such as snakes. The construction of structures with "branches" requires the use of multiple connectors per extreme. For example, the hexapod shown in Fig. 1 can be built using modules that have one connector at one extreme and three connectors on the other, i.e., no extreme of any module of the robot has more than three points of contact with other modules. The maximum number of connectors per extreme is five, as shown in Fig. 2 (Chen and Burdick[11] have also considered this type of connectors).

The actuators of a module must be strong enough to allow it to move in spite of restrictions on the power consumption and size of the actuator. Each actuator may add a degree of freedom (DOF) to the motions of the module. For example, Fig. 2 shows a module with two DOF, i.e., pitch and roll. These two DOF would allow a chain of three modules to generate all the motions necessary to place its extreme at any position and pose within the working envelope of the chain.

The sensory system of the modules is used during reconfiguration to close the feedback system that allows the detection, guidance and alignment of the modules that are in the process of docking onto each other. These sensors must be able to acquire information remotely because the modules that are docking are not in contact.

The communication system of the modules is used to allow them to collaborate with each other at both the intra-robot and inter-robot level. At the intra-robot level, the modules of the robot can use a local network based on wires. At the inter-robot level, the modules of different robots must communicate using remote links (i.e., wireless communication).

The on-board processor gives the module autonomy with respect to local processes. A fully functional processor must be accompanied by a RAM (for variables), a ROM (static tables and programs), a communication interface and the I/O ports. Again, the desired power consumption and size of the module place constraints that eliminate most off-the-shelf processors as candidates for on-board module processor.

The final hardware component needed for reconfiguration, the power supply, provides power to all the systems of the module. Unfortunately, battery technology has not scaled down at the same pace as electronics. Therefore, it is likely that the battery size will be the module component that will set the minimum size for the module. Since a single battery might not provide enough power to drive the module components, it might be necessary to use either a more powerful battery (preserve autonomy at expense to module weight) or to share the power of different modules (preserve module weight at expense of autonomy). In general, the power system is composed of the battery, the voltage regulator and the power distribution network.

The necessary capabilities of a metamorphic module also include software that must reside on the module. This software controls various actions that must be performed by the module. Among these actions, the most challenging one is probably the docking control. Each module must know how to make use of the sensing information, and how to control the modules in the robot to align two connectors to be docked.

In addition to docking, a metamorphic robot must perform other motions to accomplish regular robot jobs such as locomotion and manipulation. In contrast to classic robot control, the challenge here is to coordinate actions from multiple modules to allow the entire robot to move coherently. This must be achieved in spite of the limited motion of each individual module and the distributed nature of the control.

Finally, the metamorphic module must have the capability to recover from failures. Safety mechanisms must be built at the design stage to detect, bypass, and discard damaged modules. For example, the communication and power sharing between modules must be robust to prevent the entire system from paralyzing because of the failure of any single module.

3. IMPLEMENTATION OF ROBOT CAPABILITIES

In this section we present our approach to provide the CONRO robot with the capabilities needed for metamorphing, as described in Sect. 2. The design of the robotic module, the building block of the robot, is discussed in Sect. 3.1. In Sect. 3.2 we discuss the size of the module as a parameter used to improve the relative strength of the actuators of the module. The control mechanisms needed for locomotion and morphing are discussed in Sect. 3.3. Finally, in Sect. 3.4 we discuss the communication between modules and between robots needed to coordinate tasks.

3.1. The Robotic Module

The building block of any reconfigurable robot is the module. These modules can be heterogeneous like those of Fujita et al.,[8] designed to play very specialized roles, e.g., limbs, heads, etc. This approach has the disadvantage of requiring the individual design of each module according to its function. Another possibility is to design homogeneous modules and build the robot by assembling them in a given configuration. This is the approach taken originally to build hyper-redundant robots that move in a plane (e.g., snake robots and planar robots like those of Paap et al.,[12] Pamecha et al.,[13] and Poi et al.[14]) and later extended to locomotion in 3-D space (e.g., robotic molecule,[6] 3-D unit[15]). The use of homogeneous modules has advantages that make them suitable for modular robotics. First, the homogeneity of the modules can be used as a fault tolerant mechanism that adds redundancy to the robot. If a module fails, it can be discarded and replaced by any other module. Second, the homogeneity of the modules simplifies the mechanical design of the system. The CONRO robots are build using homogeneous modules; their behavior is assigned on the fly and depends on the role that the particular modules are playing in a particular configuration.

All the modules of the CONRO robot are structurally homogeneous and any of them can replace any other in its basic functions, i.e., docking capabilities, locomotion and communication. In Fig. 2 we show a single module of CONRO indicating some of its features. Structurally, each module has three segments connected in a chain. Two independent axis of rotation, located at the intersection of these segments, provide the module with motion capabilities. Each module has a set of connectors at the extremes of the chain, positioned as indicated by the gray areas shown on Fig. 2. Each set has five faces, each one capable of making a solid contact with any of the faces of another module. The communication between adjacent modules is performed using contact points located on the faces of the connectors.

The module uses its actuators to generate docking motions and gaits. During intra-robot reconfiguration, the actuators are used to move the face of a module toward the face of another module to dock the modules and create a solid contact between them. This is the basic step that changes the topology of the robot. During inter-robot reconfiguration, the actuators of the module are first used to generate gaits that allow the robots to approach each other and then, once the robot are close enough, the actuators are used in the docking process.

The communication mechanism allows the modules to communicate with each other for purposes of coordination of motions. Different communication systems are suitable for intra-robot and inter-robot communication. During

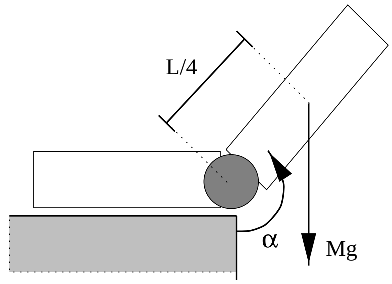

Figure 3. Simplified model of module.

intra-robot communication, the modules are in contact with each other and thus, it is possible to establish a high-bandwidth link using the connector as the point of contact between the modules. During inter-robot communication, the modules that will connect are in different robots and thus, communication must be performed using a remote link. This type of wireless communication could be performed using RF or infrared signals.

3.2. Module Size

The size of the module is selected to increase the ratio of actuator torque to module size as much as possible. We recall the physic fundamentals involved in this ratio and relate them to a simplified version of one of our modules as shown in Fig. 3. We consider two modules: module A has dimensions $L \times W \times H$ and module B, a dilated version of module A, has dimensions $Lk \times Wk \times Hk$ for $k < 1$. The volumes of module A and B are

$$V_A = LWH, \quad V_B = V_A k^3.$$

Let the modules have homogeneous density δ. Then, the masses of the modules are

$$M_A = V_A \delta, \quad M_B = V_A k^3 \delta.$$

Let the modules have only one actuator located in the middle and let one segment of the modules be fixed to the horizontal plane as shown in Fig. 3. The magnitude of the torque needed by module A to lift the free segment is

$$\tau_A = \frac{V_A \delta}{2} g \frac{L}{4} \sin(\alpha)$$

while for module B is

$$\tau_B = \frac{V_A k^3 \delta}{2} g \frac{Lk}{4} \sin(\alpha)$$

where g is the gravitational constant and α is the angle of pitch of the segment with respect to the vertical plane. Therefore, a reduction of the length of the module by a factor of k reduces the magnitude of the torque required to lift the segment by

$$\tau_B / \tau_A = k^4 \tag{1}$$

The reduction in length of the module also reduces the rotational inertia of the segment about the axis of the actuator. Let the segment approximate a thin rod. Then, the rotational inertia about the axis of the actuator is

$$I_A = \frac{M_A}{2} \frac{(L/2)^2}{3}$$

for module A while, for module B we have

$$I_B = \frac{M_A k^3}{2} \frac{(L/2)^2 k^2}{3}$$

Therefore, a reduction of the length of the module by a factor of k reduces the rotational inertia by

$$I_B / I_A = k^5 \tag{2}$$

The torque and inertia equations (1) and (2) indicate that small reductions in the size of the module translate into large reductions of torque, and therefore power consumption, i.e., as the size of the module decreases, the capability of a module to lift a chain of other similar modules increases. The equations indicate that linear variations in the size of the module affect the torque required to move the module and the inertia of the module with variations of the fourth and fifth power respectively. If we reduce the torque required to move a module, we reduce the power consumed and thus, we can reduce the size of the battery that provides it. The reduction of the size of the battery translates onto a reduction of the size and weight of the module which in turn, reduces again the requirements on the required torque. This interdependent reduction of battery size, module weight, actuator torque, and module size has implicit limits on all of these parameters, i.e., there is a limit on how small can a battery be, how small can an actuator be, how small we can miniaturize the components of the module and how light we can build the module (Wallace and Selig[16] studied the limitation of scaling for the particular case of using a direct drive motor as actuator). Our approach is to build the smallest possible module constrained by the minimization of its components, their weight, the torque required by an actuator to move such module and the battery needed to power such actuator and module components.

3.3. Robot Control

Each module is independent of the other modules, it has its own CPU and memory and it controls its own sensors and actuators. This local control has two functions. First, it provides reactive behavior to the module since, due to constraints of resources (e.g., the low bandwidth and high traffic of the network) there is no time for deliberations between modules that are "far away" in the current configuration. Second, it serves as a fault tolerance mechanism in the sense that the control of the robot does not depend on any single module. Thus, a module that fails can be disposed by the robot without affecting the overall robustness of the system. If the robot loses too many modules to failure, it might merge with another robot that is equally in need of modules to form a larger robot.

The global control of the robot needed for locomotion and morphing can be done in either a centralized or distributed way. A centralized control uses the CPU of a given module as a master computer and the CPUs of all the other modules of the robot as slaves. In this case, the robot is controlled in a manner similar to the control of a hyper-redundant robot, using partial kinematics on the chains of the robot. A distributed control does not use a master CPU. Instead, it may use a new control mechanism that is inspired by the concept of a hormone present in biological systems.

Similar to a biological system, a hormone is a control signal that travels among modules in one or more robots, triggering and coordinating their actions to accomplish complex tasks. Different types of hormones serve different purposes, triggering different modules to perform different actions. They can also serve as the synchronization signal between modules needed to produce a desirable global result. Each hormone contains a type, an action code and a set of indicators to represent and record the progress of the hormone-encoded actions. Once generated, a hormone floats through the modules just as a blood cell floats through blood vessels. Upon receiving a hormone, a module combines the action code in the hormone with the "role" of the module in the current robot configuration, and uses this information as an index to retrieve appropriate actions from reaction tables stored in the module. The module will then execute the retrieved actions (e.g., move its motor or modify the hormone's action code or progress bits), and pass the hormone to the next module. Additionally, a hormone has a life span; it "expires" when all its progress indicators are set to zero.

To illustrate the idea of hormone-based control, consider the control needed to move a chain of CONRO modules as a caterpillar using a traveling wave, e.g., a wave motion of 6 modules with bending angles of $(0, +60, -60, -60, +60, 0)$ traveling from the head to the tail of the chain. In this case, a hormone of type "c-move" is generated and released in the robot. With this hormone, the modules in the robot coordinate their actions to move the whole chain as a caterpillar.

The concept of a "hormone" is different from that of a "message" used in the conventional message-passing computer paradigm. The three major differences are: a hormone does not have a destination address, it has a life-cycle, and its content may be interpreted differently by different receivers. Because of these properties, we believe that a hormone-based control is well-suited for distributed robot control, and is general enough for both intra-robot control (internal coordination between modules) and inter-robot control (social interactions or reassembling between different reconfigurable robots).

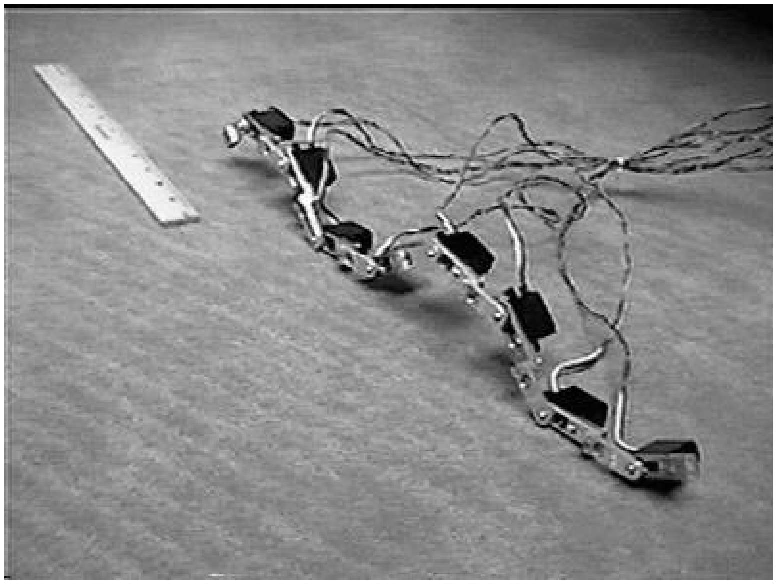

Figure 4. Snake prototype using traveling wave gait.

3.4. Communication

We are designing the modules of CONRO to be self-sufficient, each one with a local CPU and memory. Therefore, the modules of a robot (or different robots) need a communication system to coordinate their actions for purposes of morphing and locomotion.

During intra-robot operation, the communication is used for both gait coordination and module docking. For locomotion purposes, the modules communicate using a serial protocol on a local network based on wires and contacts on the faces of the connectors. In this case, the exchanged information is control signals (e.g., hormones) that coordinate the motion of the robot to achieve the gait. On the other hand, during morphing the modules that will dock onto each other need to provide some feedback information to guide each other. Since the modules are not in contact, a remote communication system is more appropriate; in our case we have selected an infrared system. In this case, the wire-based network is used to control the docking procedure while the infrared system is used to provide distance and alignment information.

During inter-robot morphing, the robots need to agree on the reconfiguration, then they must move toward each other and finally they must dock one of their modules. Since there is no direct contact between the robots until they dock, all their communication must be performed remotely. In our case, this communication will be based on the infrared system, which is suitable as long as there is a line of sight between the robots and the distances between them is not too long. Remote communication under different conditions requires radio links.

4. EXPERIMENTAL PROTOTYPES

We have build two CONRO prototypes to be used as proof of concept and test beds for the docking algorithms, the design of gaits and the development of hormones and user interfaces. These robots are tethered, controlled from a host computer running Linux and powered by an external power supply. The actuators are RC servo motors controlled by 8-axis servo controller cards. The controlling software is composed of two programs. A low level program provides a library of functions, serial port and controller drivers, synchronization routines, a motion scheduler and a trajectory generator. A high level program uses the functions to specify motions, their speeds or their duration. These two programs communicate through a UNIX socket but run independently of each other, i.e., it is possible to run the programs in different computers to take full advantage of the processor of each host or, if necessary, they can run in a single computer (e.g., for purposes of portability).

The first robot, shown in Fig. 4, is a snake formed by a chain of eight modules. The length of the robot is 515 mm and the length of each module is 70 mm. All the modules of this snake have a pitch DOF. Thus, since the modules

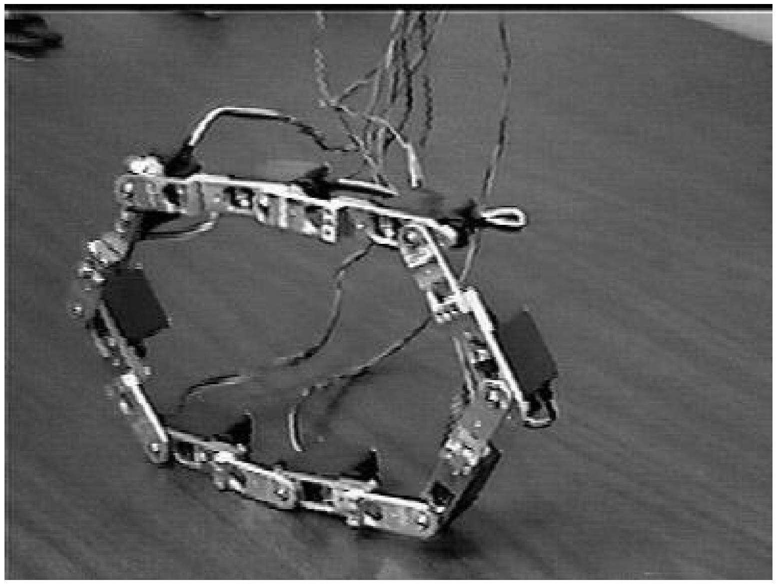

Figure 5. Snake prototype using a rolling-track gait.

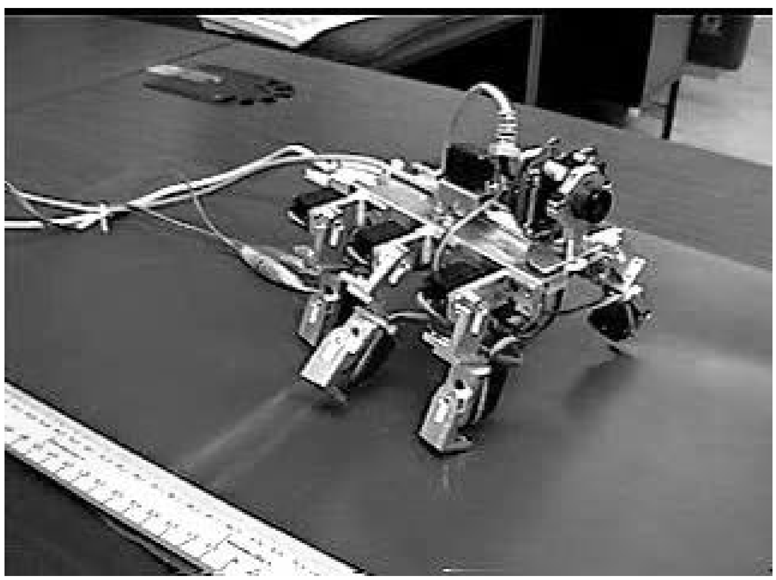

Figure 6. Hexapod prototype moving forwards.

are connected in parallel positions, this robot can only move on a plane, i.e., the robot is a hyper-redundant planar robot with eight DOF. Using hormones it is possible to execute hyper-redundant gaits, e.g., Figs. 4 and 5 show the robot performing a traveling wave and a rolling-track gait respectively similar to the robots designed by Poi et al.[14] and Yim.[17] These gaits are examples of the capabilities that a reconfigurable robot such as CONRO could achieve. The robot might use the traveling wave gait to travel on low clearance passages such as pipes or on delicate surfaces such as thin ice. If the robot needs speed and the surface is flat the robot might dock its two extremes, create a loop and use the rolling-track gait. Note that the different configurations shown in these figures were obtained by manual reconfiguration.

The second robot, shown in Fig. 6, is a hexapod whose legs are made by chains of two modules. The spine of this robot is not made out of modules but instead it is a wide platform aimed to carry an on-board camera. The length of each module is 44 mm. Notice that these modules could be reassembled manually to form a fully functional snake robot, i.e., a small version of the first prototype.

Figure 7. Legs of hexapod prototype.

All the modules of the hexapod also have a pitch DOF. However, in contrast to the snake, the modules of each leg of the hexapod are connected in orthogonal directions and thus, the movement of each leg is not confined to a plane, i.e., each leg is a non-redundant two DOF chain as shown in Fig. 7. These two DOF give each leg the mobility of a human leg but constraining the motion at the hip to the plane of the body, i.e., similar to a jumping-jack leg motion. These DOF make the motion of each leg resemble a rowing motion.

The user interface for both prototypes is based on a glove that interfaces to the computer. The direction and speed of the motions of the robots are controlled using hand gestures. The on-board camera of the hexapod provides real-time visual feedback to the user, allowing him to control the robot from a robot-centered point of view.

In parallel with the previous work, we are developing the components for the miniature module to make it self-sufficient. These components are a miniature CMOS camera, a low-power micro-processor, and miniature connectors and actuators based on shape memory alloy.

5. FUTURE WORK

Most of the work performed on the project has been devoted to designing and building the robotic modules, i.e., the hardware component. As the hardware matures, the effort will be shifted to the software component, particularly in the areas of robot control and the dynamic of morphing. The CONRO robot presents a unique test bed for distributed control because the robot itself is a network of computers. Thus, the control of a group of CONRO robots is an exercise on how to control a distributed robotic system (i.e., the group of robots) based on a different distributed robotic system (i.e., the robot itself). This control will be studied at the level of centralized control (where one module takes control of the whole system) and distributed control using hormones. We are also studying a type of control that is a compromise between centralized and distributed control, a hierarchical robot control, where a given type of control (e.g., centralized) is used to direct sets of modules that use another type of control (e.g., distributed). This type of hierarchical control is appealing because it could be applied to the control of both a single robot and a group of robots.

In addition to the control of actions, self-reconfigurable robots must know when a new configuration is needed, what the new shape should be, and how to transform themselves into the new configuration. Solutions to such problems require substantial research on integrating information from distributed sensors to form a global situation assessment, detecting and selecting a better configuration when needed, and determining a sequence of reconfiguring actions to perform the transformation efficiently. To solve these problems, we will investigate approaches at both distributed and proxy levels. At the distributed level, we will investigate the use of hormones as media for both information integration and action propagation.

6. CONCLUSION

We have introduced CONRO, a metamorphic robot targeted to real-world applications that pose intrinsic difficulties for fixed-shaped robots. In contrast to other work of metamorphic robots, the CONRO robots can separate locomotion from configuration, i.e., they can select the best configuration for a given situation and move with configuration-dependent efficient gaits. We have analyzed some necessary and desirable capabilities that a robotic module, the building block of the robots, must have and introduced the CONRO modules as a possible solution to the technical challenges raised by the desired capabilities. We have pointed out that the design and implementation of feasible metamorphic robots in terms of today's technology in actuators, sensors, processors, powers, and MEMS must take into account the size and autonomy of the modules as key concepts. The present designs and experimental results suggest that our approach may provide a feasible solution for the realization of a new class of metamorphic robots.

ACKNOWLEDGMENTS

The following people are involved in specific aspects of the CONRO project: Alberto Behar (miniature camera), Ramesh Chokkalingam (module electronics), Robert Kovac and Behrokh Khoshnevis (module connector and actuator), Wei-Hua Mao (compiler), Yi-Min Lu (user interface) and Nestoras Tzartzanis (low-power cpu). This research is being sponsored by DARPA/ETO under contract number DAAN02-98-C-4032.

REFERENCES

1. T. Fukuda and Y. Kawauchi, "Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator," in *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 662–667, 1990.
2. M. Yim, "A reconfigurable modular robot with multiple modes of locomotion," in *Proc. JSME Conference on Advanced Mechatronics*, 1993.
3. S. Murata, H. Kurokawa, and S. Kokaji, "Self-assembling machine," in *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 441–448, 1994.
4. E. Yoshida, S. Murata, K. Tomita, H. Kurokawa, and S. Kokaji, "Distributed formation control of a modular mechanical system," in *Proc. Int. Conf. Intelligent Robots and Systems*, pp. 1090–1097, 1997.
5. A. Pamecha, I. Ebert-Uphoff, and G. Chirikjian, "Useful metrics for modular robot motion planning," *IEEE Trans. Robotics Automat.* **13**, pp. 510–521, Aug. 1997.
6. K. Kotay, D. Rus, M. Vona, and C. McGray, "The self-reconfiguring robotic molecule," in *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 424–431, May 1998.
7. M. Nilsson, "Why snake robots need torsion-free joints and how to design them," in *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 412–417, 1998.
8. M. Fujita, H. Kitano, and K. Kageyama, "Reconfigurable physical agents," in *Autonomous Agents*, pp. 54–61, 1998.
9. C. Paredis and P. Khosla, "Design of modular fault tolerant manipulators," in *Proc. First Workshop Algorithmic Foundations of Robotics*, pp. 371–383, 1995.
10. B. Neville and A. Sanderson, "Tetrabot family tree: Modular synthesis of kinematic structures for parallel robotics," in *Proc. IEEE Int. Symposium on Robotics Research*, pp. 382–390, 1996.
11. I.-M. Chen and J. Burdick, "Determining task optimal modular robot assembly configurations," in *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 132–137, 1995.
12. K. Paap, M. Dehlwisch, and B. Klaassen, "GMD-Snake: A semiautonous snake-like robot," in *Distributed Autonomous Robotic Systems 2*, Springer-Verlag, Tokio, 1996.
13. A. Pamecha, C.-J. Chiang, D. Stein, and G. Chirikjian, "Design and implementation of metamorphic robots," in *Proc. 1996 ASME Design Engineering Tech. Conf. and Comput. in Engineering Conf.*, pp. 1–10, Aug. 1996.
14. G. Poi, C. Scarabeo, and B. Allota, "Travelling wave locomotion hyper-redundant mobile robot," in *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 418–423, 1998.
15. S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji, "A 3-D self-reconfigurable structure," in *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 432–439, May 1998.
16. R. Wallace and J. M. Selig, "Scaling direct drive robots," in *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 2947–2953, 1995.
17. M. Yim, *Locomotion with a unit-modular reconfigurable robot.* PhD thesis, Stanford University Dept. of Mechanical Engineering, Mar. 1994.