The Conro Modules for Reconfigurable Robots

Andres Castano, *Member, IEEE*, Alberto Behar, and Peter M. Will, *Member, IEEE*

***Abstract*—The goal of the Conro Project is to build deployable modular robots that can reconfigure into different shapes such as snakes or hexapods. Each Conro module is, itself, a robot and hence a Conro robot is actually a multirobot system. In this paper we present an overview of the Conro modules, the design approach, an overview of the mechanical and electrical systems and a discussion on size versus power requirement of the module. Each module is self-contained; it has its own processor, power supply, communication system, sensors and actuators. The modules, although self-contained, were designed to work in groups, as part of a large modular robot. We conclude the paper by describing some of the robots that we have built using the Conro modules and describing the miniature custom-made Conro camera as an example of the type of sensors that can be carried as payload by these robots.**

***Index Terms*—Autonomous, modular, reconfigurable, self-sufficient.**

I. INTRODUCTION

RECONFIGURABLE robots are modular robots that can change their shape. These robots could be used in applications that would benefit from the use of different or multiple fixed-size fixed-shape robots. A reconfigurable robot could change its shape into a snake to reach into narrow places during a rescue operation, into a hexapod to carry a load or it may split into many smaller robots to perform a task in parallel.

Reconfigurable robots are classified as homogeneous or heterogeneous depending on whether their modules are identical or not. In a homogeneous robot, the position of the module in the robot defines its function, for example, the module could play the role of head, leg or spine depending on its location in the robot. In a heterogeneous robot, the function of the module defines its position in the robot, for example, the possible positions of a leg module are restricted to the legs of the robot.

Reconfigurable robots can also be categorized according to whether or not their modules are organized in a lattice (either in the plane or 3-D space). Lattice-based robots are usually homogeneous and need to reconfigure in order to move, i.e., as their topology changes, their center of mass translates accordingly. In contrast, nonlattice robots can either translate while reconfiguring or can separate their reconfiguration and locomotion stages. This separation allows them to reconfigure and then select an efficient configuration-dependent gait.

The original reconfigurable robots were designed to add versatility to the robotic manipulator. Early work by Will and Grossman [1], Schmitz *et al.* [2], and Fukuda and Kawauchi [3] continues to evolve in the work of Paredis and Khosla [4], and others. Among the planar lattice-based cellular robots we find the robots based on square and hexagonal modules of Yoshida *et al.* [5] and Murata *et al.* [6], respectively, and the robot based on hexagons of links of Chirikjian *et al.* [7]. This work has been extended to 3-D space, among others, by the cubic units of Murata *et al.* [8], the robotic molecule of Kotay *et al.* [9], the crystal module of Rus and Vona [10] and the "I-Cube" modules of Ünsal *et al.* [11]. Most nonlattice-based reconfigurable robots are heterogeneous. Among these we find the robots for the entertainment industry of Fujita *et al.* [12] and those for space exploration of Farritor *et al.* [13].

In this paper, we describe the design approach and the mechanical and electrical aspects of the modules of the Conro robots, nonlattice homogeneous reconfigurable robots, targeted to search and rescue and surveillance operations. The Conro robots have some similarities with the Tetrobot of Hamlin and Sanderson [14]. Both are homogeneous and can separate their locomotion and reconfiguration stages. However, the Tetrobot must be tethered while the Conro robot is self-contained. Conro robots are also similar to the Polypod of Yim [15], [16] in both capabilities and concept. However, Conro robots emphasize size and autonomy as design parameters and are designed to support inter-robot reconfiguration, i.e., reconfiguration that involves more than one robot and leads to the merging of robots into a larger one or the splitting of a large robot into smaller ones.

This paper is organized as follows. In Section II we summarize the philosophy behind the design of the module. In Sections III and IV we describe the design of the module from the mechanical and electrical points of view, respectively. In Section V we discuss the considerations of the module with respect to its size. In Section VI we describe the resulting module and give some examples of the possible Conro robot configurations. In Section VII we present the miniature custom-made Conro camera built which is an example of the sensors that a Conro robot could carry as load. Finally, in Section VIII we present our conclusions.

II. PHILOSOPHY OF DESIGN

The goal of the Conro Project is to build deployable reconfigurable robots that exhibit inter-robot reconfiguration capabilities. The capabilities of these robots are determined by the characteristics and functionality of their modules. The basic shape of the Conro modules is that of three segments connected in a

---

Manuscript received March 31, 2002; revised October 2, 2002. Recommended by Guest Editor W.-M. Shen. This work was supported by the Defense Advanced Research Projects Agency (DARPA)/Microsystems Tech Office (MTO) under Grant DAAN02-98-C-4032.

A. Castano was with the Information Sciences Institute at the University of Southern California, Marina Del Rey, CA 90292 USA. He is now with the NASA/Jet Propulsion Laboratory at the California Institute of Technology, Pasadena, CA 91109 USA (e-mail: andres.castano@jpl.nasa.gov).

A. Behar is with the NASA/Jet Propulsion Laboratory at the California Institute of Technology, Pasadena, CA 91109 USA (e-mail: alberto.behar@jpl.nasa.gov).

P. M. Will is with the Information Sciences Institute at the University of Southern California, Marina Del Rey, CA 90292 USA (e-mail: will@isi.edu).

Digital Object Identifier 10.1109/TMECH.2002.806233

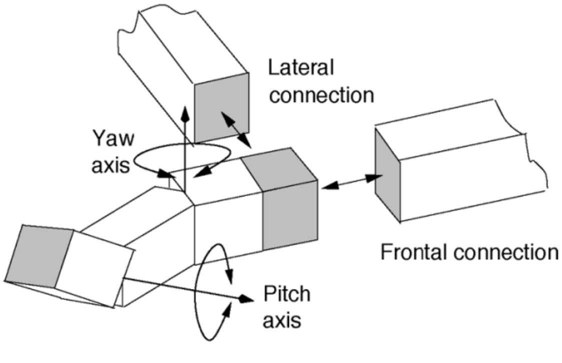


Fig. 1. Basic shape of a Conro module.

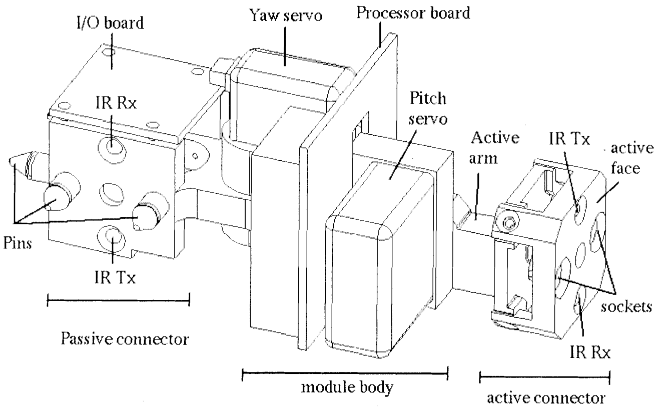


Fig. 2. Parts of the Conro module.

chain, as shown in Fig. 1. Two independent axes of rotation, located at the intersections of these segments, provide the module with motion capabilities. The extremes of each module have ports (one on each shaded face) that allows it to connect to other modules. A detailed discussion about the philosophy of design of the Conro module can be found in [17]. The summary in this section is presented for completeness of the paper.

The specifications of deployability and inter-robot reconfiguration capabilities translate into constraints on the levels of self-sufficiency, autonomy and homogeneity of the module, module size, and communication capabilities. A deployable robot must be self-sufficient, i.e., capable of untethered operation. In the trivial case, an inter-robot reconfiguration split operation may create a robot formed by a single module. Therefore, to guarantee that any robot such created is self-sufficient, each module must be self-sufficient. Likewise, a module must be autonomous with respect to the use of its own resources, for example, it has exclusive access to its sensors and actuators.

The level of homogeneity of a module determines its capabilities and the functions that it can fulfill. Each module must have a processor, power, sensors, actuators, and communication systems to satisfy the self-sufficiency and autonomy constraints. Other components not needed to satisfy these constraints (cameras, antennas, etc.) can be carried by the robot as a load or are piggy-backed on a particular module, driven by a generic interface port. This tradeoff between the necessary and desired components of a module reduces its design, manufacturing, testing and programming costs. All the components must fit into a package that is as small as possible to reduce the effect of inertia of the limbs and increase the relative torque-to-robot weight ratio of the actuators.

Finally, we address the communication needs of the module. During inter-robot reconfiguration, two robots need to communicate remotely to agree on the merging operation. Thus, robots need to exchange information remotely and need a mechanism to guide another robot toward itself. At the local level, each module needs to communicate with its adjacent modules. We concluded that an infrared-based system could satisfy all these requirements; it could be used for both remote and local communication and double as the directional guiding mechanism for both inter-robot and intra-robot dockings.

III. MECHANICAL DESIGN

Our implementation of the Conro module has three segments connected in a chain: a passive connector, a body and an active connector, as shown in Fig. 2. At the intersection of the body and the two connectors there are joints that give the module yaw and pitch degrees of freedom. The weight of the module is $W_m = 114$ g (including batteries) and its length is $L_m = 10.8$ cm excluding the length of the pins protruding from the passive connector. We now describe the parts of the module.

A. Module Body

The body is the central part of the module to which the active and passive connectors attach. It is composed of a delrin frame, two servo motors and a printed-circuit board (PCB) as shown in Fig. 2. The PCB has a hole in its center to allow its accommodation on the frame. When it is in place, the PCB is screwed in position to the frame. The servos fit into cavities of the frame and are held in place by friction.

Commercial off-the-shelf servos for radio-controlled devices (Futaba S3102 RC servos) are connected directly to the processor board using 3-pin connectors. Two of the pins provide power to the servo and one carries a pulsewidth modulated (PWM) signal that defines the position of the shaft. Each servo has a torque of $\tau_a = 3.7$ kg·cm and weighs $W_a = 21$ g, mainly because of the weight of their metal gears, i.e., the servos account for 36% of the weight of the module. The output shafts of the servos are connected directly to the active and passive connectors in a direct drive fashion. The processor board is a two-layer PCB with surface-mounted components that distributes the control signals and power to the rest of the module and serves as holding place for a small 3-V battery.

B. Passive Connector

The connectors allow the module to attach to other modules. The passive connector has no moving components. Its frame is a cube of delrin with a side of 2.54 cm as shown in Fig. 2. Three lateral faces of the cube have two protruding aluminum pins that fit into the sockets of the active connectors of other modules. The cylindrical pins have a lateral groove to allow the active connector to anchor to them. The particular positions of these pins and sockets permit only connections of modules that lie in the same plane, i.e., modules that are tilted 90° with respect to each other cannot be connected. On each of these faces there is an infrared (IR) pair used by the module for communication and docking. The fourth lateral face of the cube has a tongue that fits on a fork of the body and allows the module to pivot the connector about the yaw axis. The yaw servo is unbiased with respect to the main axis of the module, i.e., the passive connector can rotate the same angle in the right and left directions (about 60°).

The frame of the passive connector is hollow and holds the wiring of the infrared devices of the faces and the main battery of the module, a 6-V battery (9 g, 2.5-cm height, 1.3-cm diameter). The roof of the cube is a two-layer PCB that is screwed directly onto the cube. This PCB has the input-and-output (I/O) electronics that drive the infrared receivers (RX) and transmitters (TX) of the faces of the connector and doubles as the positive contact for the battery. A 14-pin connector is used to transfer the power of the main battery to the processor board and receive the control signals for the IR components. Finally, the connector has a latch at the bottom of the cube that keeps the battery in place and serves as its negative contact. The latch can swing about one of its extremes allowing the removal of the battery. The weight of the passive connector, including the battery, is 30 g.

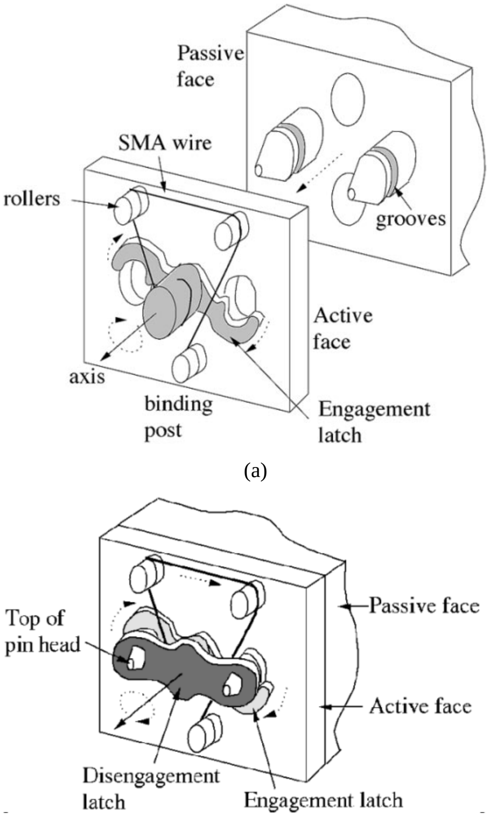
C. Active Connector

The active connector engages and disengages the pins of the passive connectors of other modules. It weighs 15 g and is composed of two parts, an arm and a face, both machined in delrin, as shown in Fig. 2. The body of the module is connected to the active connector by the arm. The active connector can rotate about a pitch axis located at the intersection of the arm and the body. The pitch servo is biased with respect to the main axis of the module; it can rotate 90° downwards but only 30° upwards. This bias allows the module to behave as the leg of a walking robot.

The face of the active connector has the same dimensions as those of the faces of the passive connector. It also has an infrared pair but the locations of the transmitter and receiver are the reverse of those of the faces of the passive connector to allow communication between modules when two modules are connected to each other.

The process of connecting two modules involves the active and passive connectors of the modules. Fig. 3(a) shows a simplified view of a passive connector approaching an active connector in a docking trajectory. The active face has two sockets to receive the pins of the passive face. As the pins slide inside the sockets, their dome-shaped heads force an engagement latch to rotate in a direction perpendicular to the trajectory of the pins. Eventually, the pins are fully inserted exposing a lateral groove into which the engagement latch edge is forced by a spring action (the spring is not shown in the figure). This docking process is completely mechanical.

The process of disconnecting two modules is initiated by the active face. As shown in Fig. 3(a), the engagement latch can be rotated using a shape-memory alloy (SMA) wire. The wire is attached between a fixed binding post and a cylinder attached to the latch. We use two rollers to establish the path of the SMA wire and to extend its working length. When the SMA is contracted, it rotates the cylinder clockwise, against a spring, retracting the latch and freeing the pins. Using this procedure we could free the pins at any moment. However, freeing the pins is not the same as disconnecting the modules. The SMA can be activated only for a fraction of a second because it consumes a large amount of power. Thus, it is possible that the modules fail to move away from each other before the SMA is de-energized, re-engaging the pins.



(a)

(b)

Fig. 3. Stages of the docking procedure. (a) Engagement. (b) Disengagement.

The disconnection process must guarantee that the modules will be free when it is finished. In Fig. 3(b) we show the faces of two already connected modules. We have added a view of the disengagement latch, a plate with two holes that, during engagement, allows the heads of the pins to go through. When the modules are connected, the edge of the engagement latch is pressed against the pins, into their grooves. To disconnect the modules, we contract the SMA wire as described before. As both latches rotate together, first the engagement latch frees the pins and then, the disengagement latch pushes the dome-shaped head of the pins out of the sockets. The distance that the latch pushes the pins is of the order of 0.125 mm. Still, this displacement is enough to guarantee that the latch will not be able to re-engage the pins when the SMA relaxes. After this process, the modules are disconnected and can be moved away from each other at any moment.

IV. ELECTRICAL DESIGN

The electrical system of the Conro module must support the control of the sensors and actuators, a communication system and a power system. The objectives of the design of the system are to minimize the number of discrete components, their overall weight and their power consumption while preserving the self-sufficiency and autonomy of the module.

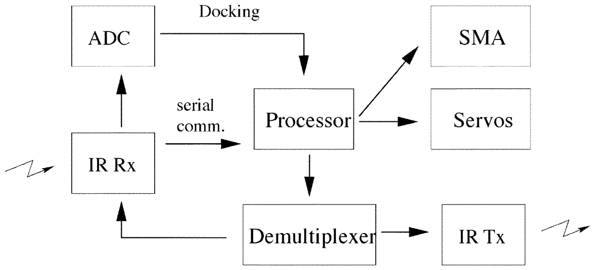


Fig. 4. Functional block diagram of a module.

A functional diagram of the electric system of the Conro module is shown in Fig. 4. Each module has a processor that gives it control over its sensors and actuators. The processor is defined by the use of one of three different single-chip microcontrollers: a stamp II based on a PIC16C57 processor or a stamp IIe or II-SX, both based on a SCENIX SX28AC/SS processor. The use of a zero-insertion force socket allows for the manual removal of the processor for replacement or programming. The three processors are pin-compatible but differ in speed, memory capacity, programming capabilities and power consumption. Thus, the processor of the module can be selected to suit a particular task according to its processor speed, memory and power requirements.

The processor has exclusive access to the actuators of the module. The SMA wire of the active connector is activated using a fixed current during a programmable period of time. The servos require a PWM signal generated in software because none of these microcontrollers has a dedicated PWM circuit. Hence, the processor must generate pulses every 20 ms to refresh the state of the servos.

The number of digital outputs of the processor was increased using a demultiplexer. Through the demultiplexer we can access both the IR receivers and transmitters of the module and establish serial communication with other modules. At this moment we can establish a 9600-baud link with/out flow control. The processor can route the input signal from the IR receiver to either a low-impedance input pin of the microcontroller or to a high-impedance input pin of an eight-bit analog-to-digital converter (ADC), depending on whether the infrared receiver is being used for serial communication or as an infrared sensor. This latter state is used during the docking of two modules, where a module uses its IR transmitter as a beacon and the other module uses its IR receiver as an analog directional sensor. The two modules that are docking can belong to the same robot or to two different robots. The combined use of these IR pairs provides the feedback necessary for the modules (or the robots) to approach each other and dock.

The Conro module uses two lithium batteries: a 6-V K28L battery, a 3-V K58L cell, each one with a capacity of 160 mA·h. The batteries set up a 9-V high-voltage low-current node to power the microcontroller and a 6-V low-voltage high-current node to power all other components. The use of the two batteries prevents large voltage drops at the microcontroller that would appear when components like the SMA or the servos are used. The batteries were selected for their voltage, size, weight, capacity, drain characteristics and the flatness of their discharge curves; lithium batteries are a good compromise between these features. Rechargable batteries, although desirable for a robotics project, have an energy density that is very inferior to the lithium chemistry.

V. CONSIDERATIONS ON THE MODULE SIZE

Specifying the parameters of the module is difficult because of their tight coupling; battery weight, motor torque and weight, module size, operating time, etc., are all parameters that affect each other. A relationship between a number of these parameters was developed during the work preliminary to the design of the modules [17]. We now apply these relationships to discuss the characteristics of the Conro module.

As described in [17], a simplified model of the module would relate these parameters with the following inequalities:

$$C_b \geq \frac{\bar{P}_m \cdot t}{V} \tag{1}$$

and

$$\tau_a \geq W_m L_m \frac{(1+2n)^2}{8}. \tag{2}$$

where

V battery voltage;
C_b battery capacity;
$\bar{P}_m$ average power consumed by the module;
t maximum operating time of the module;
τ_a actuator torque;
W_a actuator weigh;
W_m total weight of the module;
L_m length of the module;
n number of modules that a module can lift.

Equation (1) states that the battery must have a current delivery capacity C_b greater than or equal to that needed to supply the required average power $\bar{P}_m$ at the rated voltage V for a given period of time t. Equation (2) states that the torque of the actuator τ_a must be greater than or equal to that needed to handle n modules of weight W_m and length L_m or, equivalently, assuming that the actuator is located at the center of the module, the torque needed to overcome the inertia of $(2n+1)/2$ half modules, each with a weight of $W_m/2$ and length of $L_m/2$. This is the maximum torque that the actuator might need to deliver continuously.

We can use (1) and (2) to estimate upper bounds of the Conro module on n and t. The average power consumed under load by the CPU (i.e., 20 mA at 9 V), other electronics (i.e., 130 mA at 6 V) and each actuator (i.e., 150 mA at 6 V) are 180 mw·h, 780 mw·h, and 900 mw·h, respectively, so the average power consumed by the module (using only one actuator) is $\bar{P}_m$ =1860 mw-h. The equivalent battery of the module has a capacity of $C_b = 160$ mA · h and is rated at a voltage of $V =$ 9 V. Given an actuator torque of $\tau_a = 3.7$ kg·cm, we find from (1) that

$$t \leq \frac{C_b V}{\bar{P}} = 0.77\,\mathrm{h}$$

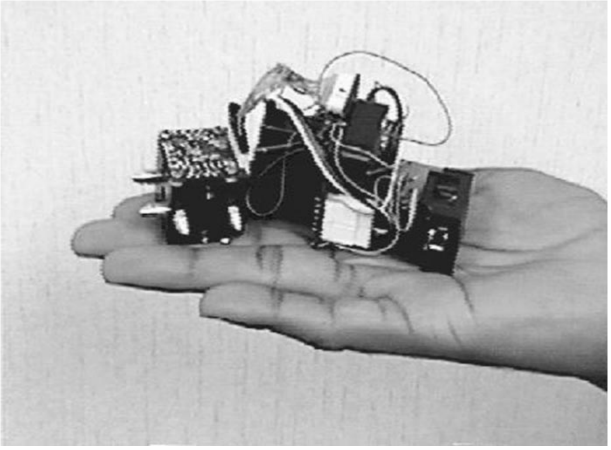


Fig. 5. Self-sufficient Conro module in motion.

and from (2) that

$$n \leq \sqrt{\frac{2\tau_a}{W_m L_m}} - \frac{1}{2} = 1.95$$

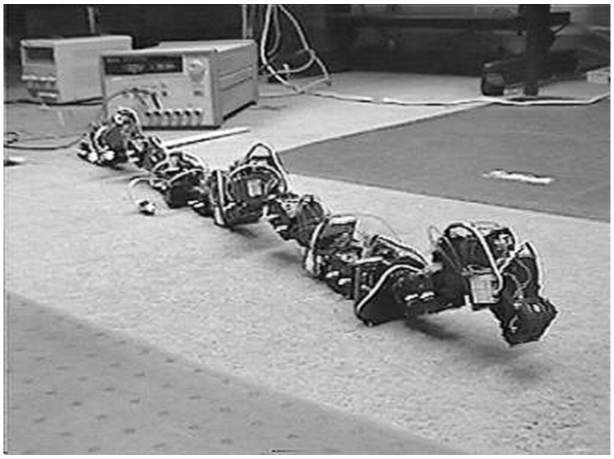
Thus, our module cannot work more than 45 min and cannot lift more that one identical module. These estimates agree with our experience with the module, for example, continuous operation of $t \approx 35$ min and $n \approx 1$ module.

VI. CONRO MODULE

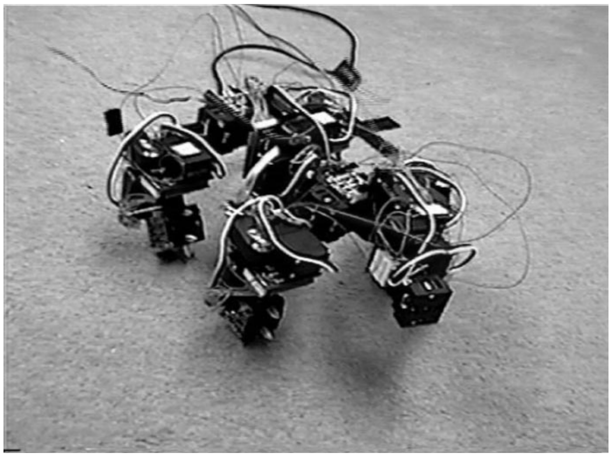
We have built twenty modules that follow the description discussed in this paper. Because each module is self-contained and autonomous, it is a robot in its own right and thus, it is possible to program it to execute motions and to react to stimuli. Fig. 5 shows the module, completely untethered, running a program that rotates its connectors in sequence. Using the PIC-based microcontroller, the program can run for 35 min continuously. Due to the high cost of running experiments using nonrechargable batteries, our robots are tested using external power supplies.

Although the modules can be run by themselves, they were designed to work in groups, connected to each other forming large robots. At this moment, the robots are configured manually, but our goal is to achieve automatic robot reconfiguration. The priority of the modules of a robot is to communicate efficiently with their adjacent modules. The modules do not share a clock signal and thus, robot actions that require the synchronized motions of different modules rely on the quality of the communication. The programming of the communication network is complex because, due to the lack of interrupt mechanisms in our microcontrollers, the module has to poll the ports in a round-robin fashion. At this moment, the infrared communication between adjacent modules is a 9600-baud inverted serial connection with flow control. The length, format and contents of the messages depend on the specific type of control used to command the robot.

The control of a Conro robot can be performed using a distributed control [18], a centralized control based on a master-slave hierarchy or a hybrid combination of both schemes. The robots shown in Fig. 6(a) and (b) are controlled using a master-slave approach where the master is a remote host with a large computational capability running C/C++ code under Linux. In this particular case the messages that we are using are three bytes long and contain information about the source and destination of the message, a message identification tag and a command token along with its respective argument. There is no network description; instead, messages intended for a particular module are broadcasted and hop through the network until they reach their destination. Fig. 6(a) shows an eight-module Conro snake executing a traveling wave gait (for example, see [19]). The particular configuration of the snake simplifies the communication mechanism because each module needs to communicate to, at most, two other modules. Fig. 6(b) shows a six-module quadruped in the middle of a walk. The large number of degrees of freedom of the robot allows for a variety of 4-legged gaits. The quadruped has been programed to move using only its legs, swinging its spine from side-to-side and a combination of both.



(a)



(b)

Fig. 6. Conro robot. (a) Snake. (b) Quadruped.

Fig. 7 shows a team of two Conro hexapods. Each hexapod is composed of nine modules: six modules play the role of legs and three modules form the spine. In this case, the distribution of the communication load is uneven because the modules that form the spine have to handle up to four communication ports (a spine module might be attached to up to four adjacent modules) while the modules that form the legs need to handle a single

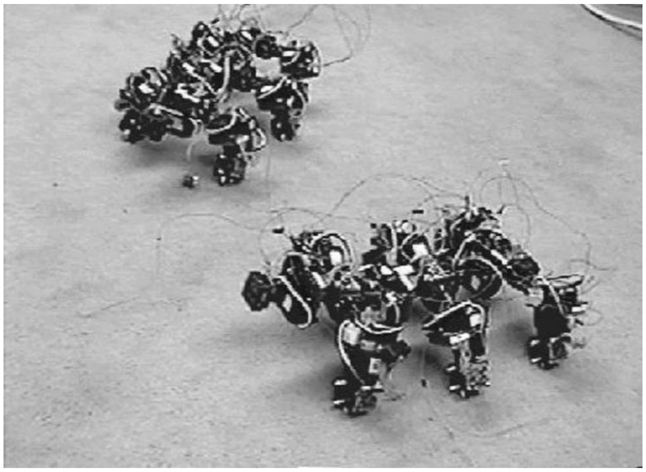


Fig. 7. Team of Conro hexapods.

port (each leg is attached to a single "spine" module). Presently, the hexapod Conro robots are capable of standing up, walking, turning, and lying down, all by their own means.

One of the aspects of controlling these robots is providing a user interface to allow an operator to interact with the environment. Toward this goal we have developed the Conro camera that can be mounted on the module that would work naturally as the "head" of the robot.

VII. CONRO CMOS DIGITAL CAMERA

As discussed before, the sensors carried by every module are those that we consider necessary to assure its self-sufficiency and autonomy, for example, the IR receivers. Still, some sensors that do not need to be carried by every module are necessary. These sensors must be small and light and are either piggy-backed on a particular module or the robot carries them as a load. In this section, we describe our work on producing one such sensor, the Conro-CMOS digital camera, that can be carried by a module as a piggy-back device. The success of this design indicates that many other relevant sensors that would fit within the tight constraints of the basic module could be produced with a focused effort; compasses, wireless links, etc.

Sensors and actuators carried by a Conro module or robot need to be as self-contained as possible in terms of memory, computational needs and power. Ideally, these sensors (or actuators) should work independently from the module and exchange information with the processor only when required. Likewise, if possible, they should carry their own batteries, memory and processing circuitry, i.e., they should not drain module resources. These goals are not always attainable but nonetheless, they should be taken into account in the design of these devices.

The Conro camera, shown in Fig. 8, was designed to be as self-contained as possible to minimize interaction with the resources of the module. It is small ($16 \times 16 \times 13$ mm^3), light (2.7 g) and computationally self-sufficient. Indeed, the camera, based on the VV5300 low-resolution digital CMOS image sensor chip produced by VVL, does not require external circuitry: the VVL chip is mounted on a board together with a 10-MHz clock, an EEPROM to store the camera startup configuration and the circuitry to drive the chip and sense its pixel array. A bidirectional 2-wire serial communications interface allows the device to be configured and its operating status monitored. Seven additional cables are used to interface the camera and the driving processor. Furthermore, the VVL chip has automatic gain control which allowed us to use a single fixed-aperture lens. With our present set up, the camera provides images with a field of view of 12°.

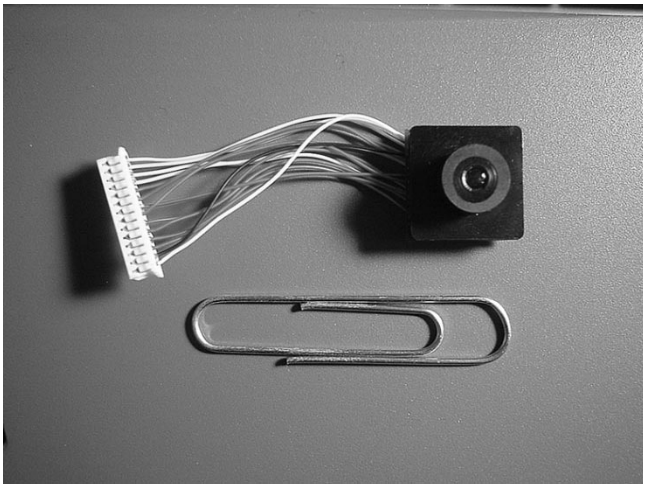


Fig. 8. Conro CMOS camera.

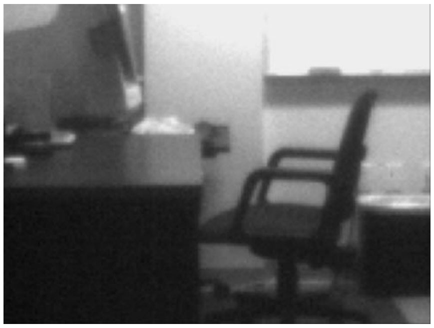


Fig. 9. Sample picture.

We have built color and monochrome versions of the Conro camera. They provide an eight-bit video stream at 30 frames/s; we can change this rate using an on-board clock divider. Individual 164×124 raw-format images can be easily obtained from the stream. Fig. 9 shows the image of a room obtained with the Conro monochrome camera.

Currently, the Conro module can carry the camera but it does not have the computational resources to capture an image (for example, the memory of the module is 2 kB while the image size is 22 kB) so instead, the camera is interfaced directly into the serial port of a PC. This is possible because the camera is computationally self-contained, i.e., the on-board oscillator and boot EEPROM provide the means to use the camera as a stand-alone unit, i.e., it can be used for the navigation of Conro robots or work as a standard PC video camera.

VIII. CONCLUSION

We have introduced the Conro module, the basic block of a reconfigurable robot that is designed for deployability and inter-robot reconfiguration. It is, to our knowledge, the first module for reconfigurable robots that is itself a robot, i.e., it is self-contained and autonomous.

The Conro modules are homogeneous, self-contained, autonomous, miniature and use an IR system for communication that doubles as a tracking system. We have described the mechanical and electrical aspects of the module and have emphasized the importance of the size of the module as an important aspect of the design and show how it relates to other module design parameters. Finally, we have described the use of sets of modules to build complex multirobot systems such as modular snakes and hexapods, each requiring the coordination of many independent robots (i.e., the modules) to operate. These robots are currently operational, i.e., they can stand up and walk, change directions and perform any other action that a similar nonreconfigurable robot of the same topology can perform (please, see http://www.isi.edu/conro for pictures and movies of the experiments). These robots are already being used as research platforms for both distributed and centralized robot control.

ACKNOWLEDGMENT

The authors would like to thank R. Chokkalingam, R. Kovac, and B. Khoshnevis for their contributions to the development of the module.

REFERENCES

[1] P. Will and D. Grossman, “An experimental system for computer controlled mechanical assembly,” *IEEE Trans. Comput.*, vol. C-24, pp. 879–888, Sept. 1975.

[2] D. Schmitz, P. Khosla, and T. Kanade, “The CMU reconfigurable modular manipulator system,” Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-88-07, 1988.

[3] 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 Automation*, 1990, pp. 662–667.

[4] C. Paredis and P. Khosla, “Design of modular fault tolerant manipulators,” in *Proc. 1st Workshop Algorithmic Foundations Robotics*, 1995, pp. 371–383.

[5] E. Yoshida, S. Murata, S. Kokaji, K. Tomita, and H. Kurokawa, “Micro self-reconfigurable robotic system using shape memory alloy,” in *Proc. Distributed Autonomous Robotic Systems*, 2000, pp. 145–155.

[6] S. Murata, H. Kurokawa, and S. Kokaji, “Self-assembling machine,” in *Proc. IEEE Int. Conf. Robotics Automation*, 1994, pp. 441–448.

[7] G. Chirikjian, A. Pamecha, and I. Ebert-Uphoff, “Evaluating efficiency of self-reconfiguration in a class of modular robots,” *J. Robot. Syst.*, vol. 13, pp. 317–338, Aug. 1996.

[8] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji, “A 3-D self-reconfigurable structure,” in *Proc. IEEE Int. Conf. Robotics Automation*, May 1998, pp. 432–439.

[9] K. Kotay, D. Rus, M. Vona, and C. McGray, “The self-reconfiguring robotic molecule,” in *Proc. IEEE Int. Conf. Robotics Automation*, May 1998, pp. 424–431.

[10] D. Rus and M. Vona, “A basis for self-reconfiguring robots using crystal modules,” in *Proc. IEEE/RSJ Int. Conf. Intellegent Robots Systems*, Oct. 2000, pp. 2194–2202.

[11] C. Ünsal, H. Kiliççöte, M. Patton, and P. Khosla, “Motion planning for a modular self-reconfiguring robotic system,” in *Proc. Distributed Autonomous Robotic Systems*, 2000, pp. 165–175.

[12] M. Fujita, H. Kitano, and K. Kageyama, “Reconfigurable physical agents,” *Auton. Agents*, pp. 54–61, 1998.

[13] S. Farritor, S. Dubowsky, N. Rutman, and J. Cole, “A systems-level modular design approach to field robotics,” in *Proc. IEEE Int. Conf. Robotics Automation*, 1996, pp. 2890–2895.

[14] G. Hamlin and A. C. Sanderson, *Tetrobot: A Modular Aproach to Reconfigurable Parallel Robotics*. Norwell, MA: Kluwer, 1998.

[15] M. Yim, “A reconfigurable modular robot with multiple modes of locomotion,” in *Proc. JSME Conf. Advanced Mechatronics*, 1993, pp. 283–288.

[16] ——, “Locomotion with a unit-modular reconfigurable robot,” Ph.D. dissertation, Stanford Univ., Dept. Mech. Eng., Stanford, CA, 1994.

[17] A. Castano, W.-M. Shen, and P. Will, “CONRO: Toward deployable robots with inter-robot metamorphic capabilities,” *J. Auton. Robot.*, vol. C-24, pp. 879–888, May 2000.

[18] W.-M. Shen, B. Salemi, and P. Will, “Hormone for self-reconfigurable robots,” in *Proc. Int. Conf. Intelligent Autonomous Systems*, 2000, pp. 918–925.

[19] G. Poi, C. Scarabeo, and B. Allota, “Travelling wave locomotion hyper-redundant mobile robot,” in *Proc. IEEE Int. Conf. Robotics Automation*, 1998, pp. 418–423.

Andres Castano received the B.S. and M.E.E. degrees from the University of Los Andes, Bogota, Colombia, in 1987 and 1989, respectively, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign, in 1992 and 1998, respectively, all in electrical engineering.

Previously, he worked at the Beckman Institute, University of Illinois at Urbana-Champaign in the area of visual servocontrol of manipulators and panoramic cameras with very large depth-of-field. He then became the Lead Roboticist of the Conro Project on reconfigurable robots at the Information Sciences Institute, University of Southern California, Marina Del Rey. Currently, he is with the NASA/Jet Propulsion Laboratory at the California Institute of Technology, Pasadena. His research interests are robotics, computer vision, and graph theory.

Alberto Behar received the B.S. degree from the University of Florida, Gainesville, in 1990, the M.E. degree from Rensselaer Polytechnic Institute, Troy, NY, in 1992, the M.S. degree in computer science (specializing in robotics) and the Ph.D. degree in electrical engineering from the Univeristy of Southern California, Los Angeles, in 1994 and 1998, respectively.

Currently, he is a Member of the Robotic Vehicles Group, NASA/Jet Propulsion Laboratory, California Institute of Technology, Pasadena, where his group designs the rovers and *in-situ* surface systems for several planetary missions. He and his group are currently designing a miniature rover to traverse the asteroid Nereus in September of 2003. His primary research interests are in sensor systems and architectures for mobile robotics and planetary surface spacecraft.

Peter M. Will received the B.Sc. degree in electrical engineering and the Ph.D. degree in nonlinear control systems from the University of Aberdeen, Aberdeen, Scotland, U.K., in 1958 and 1960, respectively.

Previously, he spent 16 years at the IBM T. J. Watson Research Laboratories, Yorktown, NY, seven years with Schlumberger, Houston, TX, five years at Hewlett-Packard Labs, Palo Alto, CA, and six years as a member of the Information Science and Technology (ISAT) Group working with the Defense Advanced Research Projects Agency (DARPA). Currently, he is the Director of the Distributed Scalable Systems Division at the Information Sciences Institute, University of Southern California (USC), Marina Del Rey, and a Research Professor of industrial and systems engineering at the same university. He has over 50 publications and 10 patents.

Dr. Will was awarded the International Engelberger Prize in robotics in 1990.