Proof of Concept of a Hyper Redundent Reconfigurable Modular Manipulator System

Timothy Vittor
Richard Willgoss
Javaid Iqbal
University of New South Wales
Sydney, NSW 2052
Australia
t.vittor@student.unsw.edu.au; r.willgoss@unsw.edu.au; jiqbal-EME@nust.edu.pk

Abstract

We present the results of a proof of concept implementation of a Hyper-Redundent Reconfigurable Modular Manipulator system (RMMS) in an attempt to validate the Modular Decentralized Control (MDC) technique in a real world environment. Hyper-redundancy enables flexibility in configuration space. Decentralized control of manipulators enables modular construction with its attendant ease of design and reduced cost. The need for a full environment reconstruction in one master controller is bypassed by having localized and simpler sub-goals for each module. The manipulator has been designed with each clone module a self-contained structure including mechanical hardware, actuators, microprocessors and motor controllers. Flange interfaces at both ends of a module allowed for ease in system reconfigurability. Each module had two rotational DOFs and formed part of an exemplar serial manipulator allowing for 3D end effector position and orientation control as determined from a combination of all module's local actions.

Test data from the environment in real time was transmitted to the first module via a stereo vision system. Each module in the system then took pseudo-independant control action, communicating through receiving and transmitting data between adjacent serial modules that formed a robot anthropic structure. The system also had the capability to avoid obstacles in real time by inclusion of bump sensors on each module. The results demonstrated the predicted fault-tolerant, robust nature of the control approach, achieving system flexibility and versatility. The MDC method is capable of extending across an extensive network of interconnected modules that would be limited more by mechanical configuration loading limitations, not communication issues. MDC has a full near world-wide patent covering its intellectual property.

1 Introduction

Manipulator systems, to date, have been customized and then optimized for high-speed repeatable motion and increasing accuracy. These capabilities have advantages in controlled operating environments, but for manipulators to become more effective in unstructured environments, they need to actively exploit redundancy to give sensible motion in negotiating around obstacles, perhaps at the sacrifice of speed. Applications that benefit from the motion of such manipulators can be as diverse as moving inside nuclear reactors [Mclean and Cameron, 2003; William II and Mayhew IV, 1997; Marzwell and Slifko, 1995; Mavroidas, et al., 1995] to movement in amongst trees to pick fruits in automated harvesting [Sarig, 1993; Hayashi, et al., 2002; Henton, et al., 2002]. In recent years, manipulator usage has been expanded to aid in service applications such as hospitals and aged or disabled care. The design, structure and control of manipulators, currently customized for the mass production market, is now developing to include system redundancy and modularity.

The shift to flexible redundant hardware configurations has become feasible with the implementation of rapidly deployable manipulator systems [Paredis et al., 1996] designed and constructed as Reconfigurable Modular Manipulators (RMM). RMM systems are also capable of developing their own inverse kinematics using real time algorithms [Kelmar and Khosia, 1990; Schmitz, et al., 1989; Kelmar and Khosia, 1988] once a system configuration is decided upon. Another modular system, including its architecture [Chen, et al, 1998] and implementation [Chen, 2000], has been realised in an rapidly deployable RMMS. Each workcell was based on component technology, modularising hardware, software and control. The plug and play components were modelled using a product of exponentials [Chen & Yang 1996] formulation during assembly, allowing a module set to work outside a stereotypical manipulator configuration i.e. a combination of SCARA and articulated robotics.

However, the use of multiple processors, working in a modular framework, but still communicating with a central processor, presents real-time co-ordination problems [Yamakita, et al., 2003; Matsuno and Suenaga, 2003] constraining system controllability. For increased redundancy, the centralised modeling and design procedures lead to more complex algorithms that increase computational delays, again complicating the system's controllability. To control redundant reconfigurable systems, it has been found that decentralized control architectures are more desirable [Yook, et al., 1997] because they distribute decision making into simpler and faster components.

The Modular Decentralised Control (MDC) approach utilizes a physical modular link layout of a redundant manipulator allowing multiple link-embedded processors to work to achieve a common goal. Each module or link is usually a clone of a stereotype. This approach has potentially enabled full control of redundancy, use of an arbitrary number of modules and incremental degradation in that a system will still achieve a common goal even when modules fail and are not replaced.

The stability and solution space of the MDC approach [Vittor, et al., 2003; Vittor, et al., 2004] has been investigated in order to predict end effector motion up to an n-module hyper-redundant manipulator. Previous work [Vittor and Willgoss, 2005a] has created a stability analysis for motion of a simple MDC-RMM two-module system. A modified root locus technique was invented to describe how the system's stability, expressed in classical terms, changed as a function of path traversed, brought about by the pseudo-independent actions of each module. Results showed a system that was ultimately stable, passing through unstable regions of motion yet not displaying the usual symptoms of such motion i.e. no oscillatory or bizarre motion. Further work [Vittor and Willgoss, 2005b] has now extended this stability analysis to a general n-module system.

This paper summarises an attempt to test the MDC hypothesis by designing and building a modular manipulator system to be a real world embodiment of the proposed MDC approach in manipulator control. The manipulator was designed with each module being a self-contained structure including mechanical hardware, actuators and sensors, microprocessor and motor controllers. We report the proof of concept implementation of the MDC methodology, demonstrating goal target tracking, obstacle avoidance and fault tolerance.

2 Modular Reconfigurable Manipulator System

The manipulator design was tailored to be a reconfigurable system and built on the premise that system flexibility was to be obtained by independent control of modules with limited interaction. The modular reconfigurable manipulator exemplar system is shown in Figure 1, a serial manipulator. Each module comprised the following four major sub-systems: mechanical hardware, software, electronics and actuation. Each of these sub-systems was again designed for modularity with ease of assembly and disassembly.

To create real time, real world data, a stereo vision camera system was devised and mounted on the free end module of the manipulator. This allowed the end effecter to determine the 3-dimensional displacement between the desired position and the current position of the end effecter. This information was broadcasted through the modules as the primary data for exercising decentralized control. Proximity sensors were mounted on the external casing of each module, allowing for the individual modules to detect any adjacent objects and take local avoidance action.

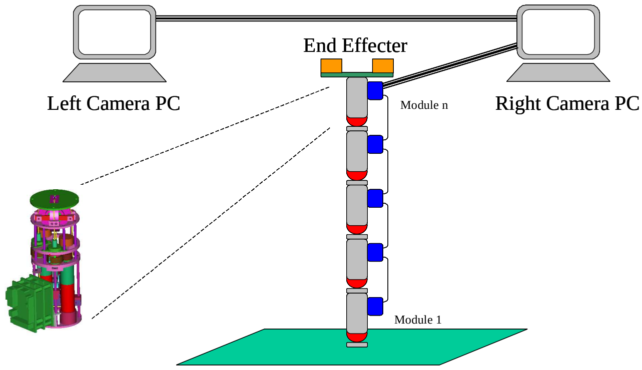

Figure 1. Reconfigurable Modular Manipulator System Layout

As mentioned, the primary system input occurred at the stereo cameras, attached to their respective image processors. A resultant x,y,z coordinate of the goal was passed to the top module, which passed it on to a subsequent module. This process continued from module to module through the manipulator, each time the local code transforming coordinates to the local cartesion x,y,z frame of reference. The required interfaces for each module included a UART and SPIM/SPIS communication, proximity sensors, hardware flanges and electrical power.

3 Stereo Vision

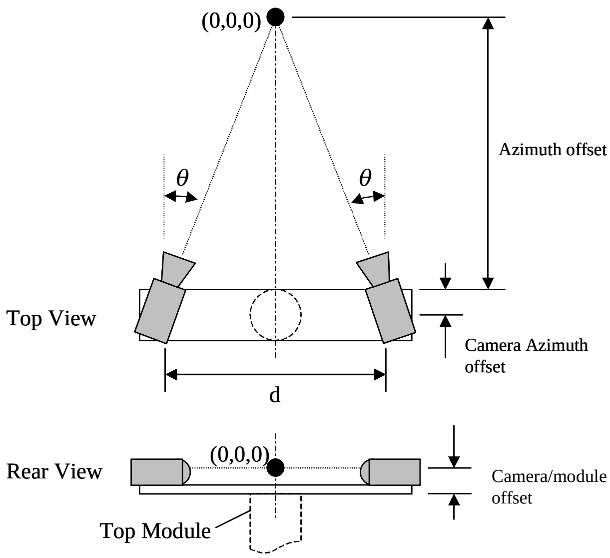

Figure 2. Stereo Vision Camera Setup

Each camera was connected to a frame grabber that provided image processing to identify its respective 2D goal position. Both frame grabbers combined their data to calculate the spatial co-ordinates of the goal in the end effecter's frame of reference. The setup of the camera limited the goal position to be initially placed within a limited +/-70mm box from the end effecter origin position to be in the field of view. A goal offset virtual link was introduced, such that the end effecter settled on a goal placed at position (0,0,0) as shown in Figure 2. The respective 2D goal co-ordinates (x_l, y_l, x_r, y_r) were found as required by the top module in its respective frame of reference, determined via homogeneous transformation.

4. Hardware

The primary hardware specifications of the modules required a lightweight, cost effective, self-contained and robust link, capable of a universal 2 DOF rotation. The initial manipulator design specifications are shown in Table 1.

Module Parameter	Value
Rotational DOF	2
Length / Diameter	205mm / 76mm
Mass	1.2 kg
Angular Range	+/- 20 Degrees
Max. Angular Velocity	20 Degrees/s
Max. Angular Acceleration	50 Degrees/s^2

Table 1. Manipulator Module Properties

The module mechanical design was divided into five primary sub-assemblies: frame, joint, drive mechanism, pulleys and casings. The module assembly, as shown in Figure 3, was designed to be assembled at minimum cost and from readily available materials and to be interchangeable in the configuration if required i.e. all clones.

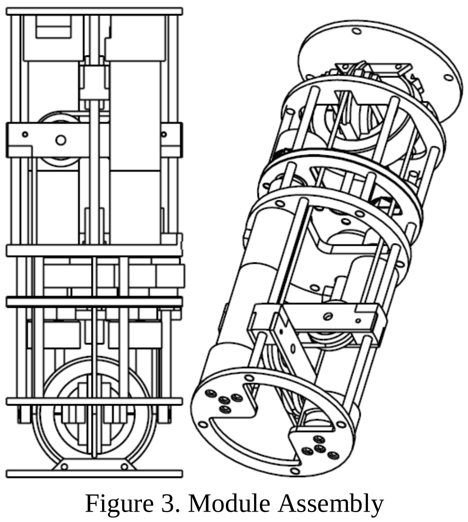

Figure 3. Module Assembly

4.1 Frame

The frame sub-assembly formed the skeleton of the module providing its primary strength and platform for mounting the remaining sub-assemblies. The essence of the design of the frame consisted of four aluminium discs connected using stainless steel threaded rods. The module frame formed a cylindrical structure 175mm in length and diameter 76mm and weighed approximately 120 grams. A working prototype would take this design to the next stage and make it out of carbon fibre composite.

4.2 Joint

The joint was designed to enable interfacing two adjacent frame structures and, at the same time, required an integration and tradeoff of closely coupled capabilities like joint strength, rotational angle range, 2 DOF rotation, compact design and light weight, most of which normally conflict. The joint consisted of two orthogonal ball bearing pairs on sleeves, one pair mounted inside the other and driven independantly, providing enough strength to support at least 5 modules, each estimated to be of order 1000gm each. The joint alone weighed 170 grams. The joint design has been formally patented and is current in Australasia, USA, Canada and Europe. A further design exercise on the joint may remove approximately 50g from its current weight without reducing its load carrying capacity.

The joint rotation was controlled by two steel cables, one attached to the outer sleeve providing rotation via the outer bearing pair and the other attached to the inner sleeve, providing rotation to the inner orthogonal bearing pair. The cable attachments allowed bi-directional control of each joint's DOF, minimizing the torsional strength requirements of the joint. The cable could sustain up to 2kN of tension, permitting torque loads of up to 50Nm. Once the joint was assembled, all the parts were locked in place using high strength steel dowel pins, ensuring rigidity of the joint.

4.3 Drive Mechanism

There were two identical drive mechanisms within each module, offset by 90 degrees, allowing control of the two-rotational DOF joint. Actuation was achieved by a brushless three phase motor and attached planetary gear head. The planetary gear then transmitted torque to a lead screw in a linear ball screw actuator. An optical encoder measured rotational angle of the gear head, providing angular positional feedback to the controller. The ball screw motion was fully constrained in rotation such that it transformed all rotational input to translational actuation.

4.4 Pulleys

Two pulley sub-assemblies were used within each module in a continuous loop arrangement and maintained the steel cable's minimum curvature requirements. Pulleys were introduced to allow for bi-directional control of the joint using a single cable drive. The pulley sub-assemblies provided for pre-tension on the steel cables, accommodating for any dimensional discrepancies in the module assembly, thereby minimising joint backlash. Use of pulleys maximized the joint angle range, allowing for a more compact design to be possible. The weight of a single set of pulleys was 100 grams.

4.5 Casing

The modules were enclosed in casings to provide housing for the module components and enable local obstacle proximity sensing. In this original design, the expense of designing all the electronics to sit completely inside the module was obviated due to the high cost involved. As the module's electronics were attached to the side of the module, the casings were designed to accommodate for their protrusion. The casing consisted of four cylindrical quadrant leaves, designed for ease of attachement or removal using spring clamps onto the module frame. The leaves sat on silicone spring-mounts and acted as touch sensors for the module through micro switches mounted onto the module frame. A combination of closed microswitches located a particular touch position.

4.6 Assembled Hardware

In the current design, to demonstrate modularity, module size and capabilities were kept as clones throughout the manipulator. In future, scaling the size of these modules would be possible such that the base module would be largest in size and power capability and modules moving away from the base module would be reduced accordingly. The assembled manipulator is shown with all covers removed in Figure 4 as a serial anthropic arm.

Figure 4. Assembled Manipulator Hardware

5 Software

The system's software was written using a Cypress Microsystems PSoC Designer platform. All module code was written using the C language. The speed of the micro controller clock was set to 12 MHz. The software consisted of the following functions, with their corresponding interfaces shown in Figure 5.

- Drive Motor
- Read Encoder
- Read Proximity Sensor
- Kinematic Transformation
- UART and SPIM/SPIS Communication

The main code routine operated by reading the optical encoders, driving motors while polling the SPIS/SPIM and/or UART communications. Each module's software was written such that the module continuously moved towards its current local goal, this being updated through the network. Despite care with electrical design, errors in the SPIM/SPIS data occurred from time to time. The polling method enabled a natural filtering of erroneous data in that extensive syntax checks discarded informal packages. The refresh rate was fast enough for the loss of an update to be almost un-noticeable in any continuous motion control.

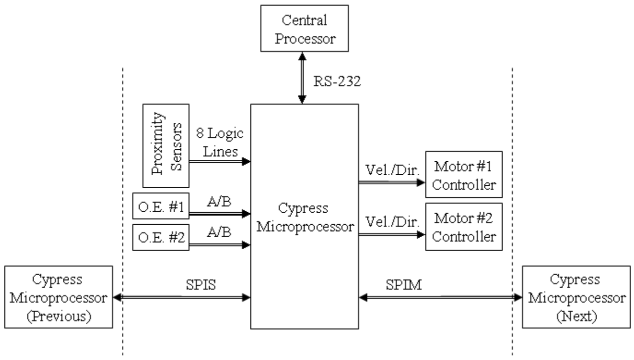

Figure 5. Module Software Interfaces

It was found that a software duty cycle of 0.1 seconds provided sufficient networking capabilities to demonstrate the algorithm in a working manipulator. The kinematic transformations were found to consume the most significant software run time. As it was desired that the code be written with as fast a polling time as practicable, two options were considered. Firstly, to speed up kinematic transformations, lookup tables were implemented. This, however, turned out to be not a feasible option as it consumed much available memory space without providing a sufficiently high angular resolution. The second option employed calculated a reduced set of iteration-dependant trigonometrical functions only and enabled goal position via the local kinematic transformation, equation 1, to be calculated within the proposed polling cycle.

$$\begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix} = \begin{bmatrix} C_1C_2 & -S_1 & -C_1S_2 & l_1 \\ S_1C_2 & C_1 & -S_1S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (1)$$

Two software codes were produced respectively for the two possible modes of control of modules:

- UART/SPIM module communicating with cameras via a RS 232 port and another module.
- SPIM/SPIS module communicating between adjacent modules.

Communication synchronization interrupts were not used to prevent disruption to time-critical software routines. As the SPI communication required specific byte synchronization, slave devices had to complete all routines and returns prior to the master device initiating new data into the pipeline. The first time-critical software routine set motor speed proportional to the magnitude of the last local goal angle errors. The second time-critical software routine read encoders to calculate the module's new joint angle errors. There was no room for quadrature counting hardware so motor direction had to be retained via software. The manipulator network, flow of data and independent motion of each module was monitored and is shown in Figure 6.

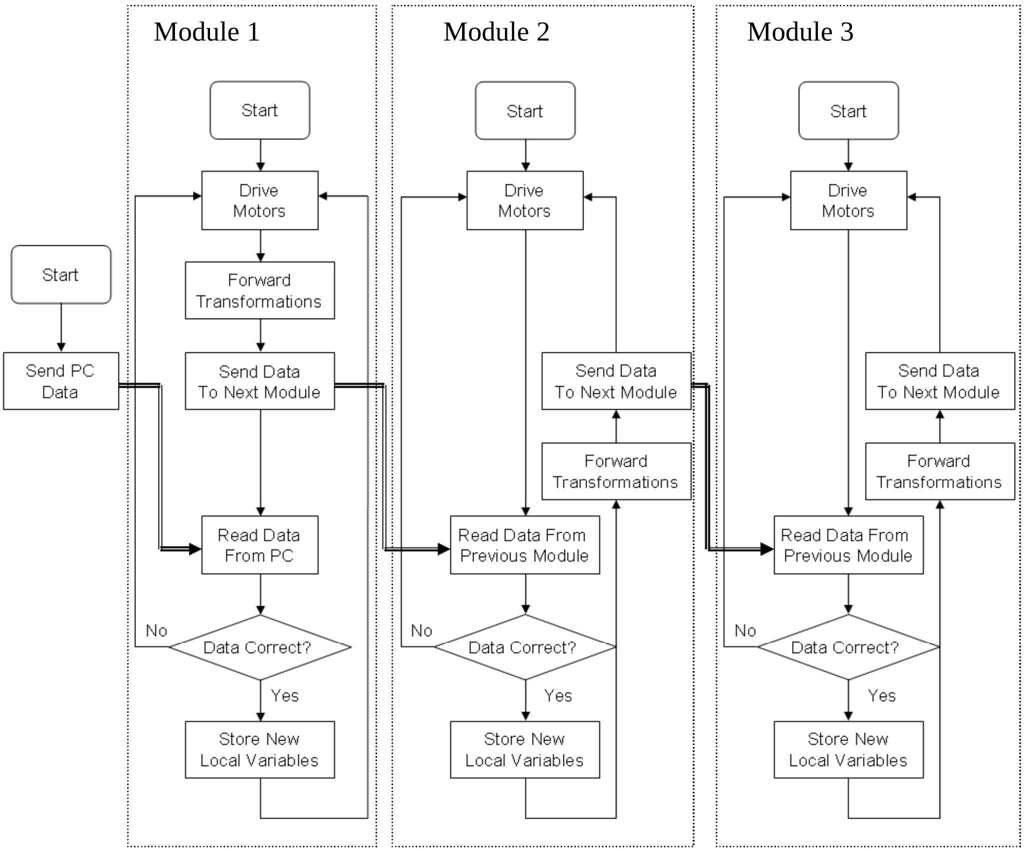

Figure 6. UART/SPIM Software Flow Chart

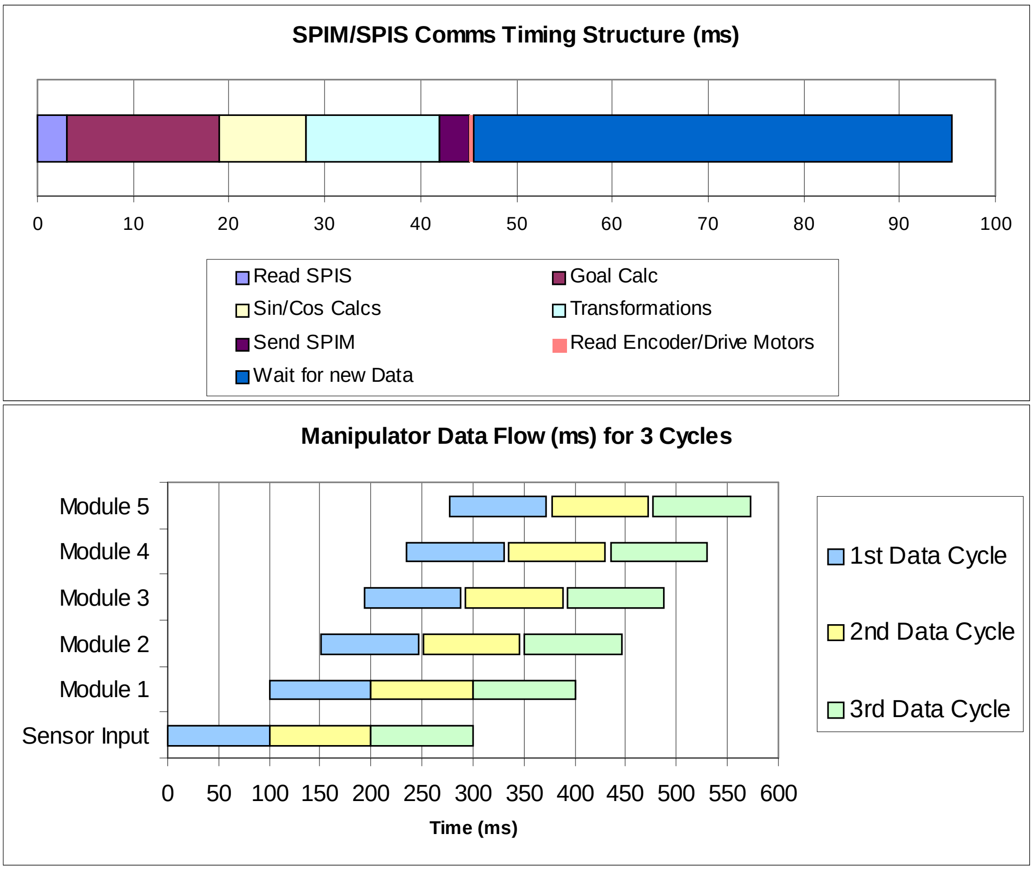

Figure 7. Module Microprocessor Data Flow

The software-sequenced approach provided a dynamically scalable network, demonstrating the effect of module addition/removal for a multi module manipulator. The duty cycles for the respective two software codes were timed to further understand the data flow within a real time system. The transformations alone consumed approximately 50% of the software run time, while the goal calculations consumed the second highest run time at approximately 30%. The lowest run time function included reading and interpreting optical encoders and controlling the motors, using only 1%. Based on these current duty cycles within the modules, Figure 7 shows the data flow within the whole manipulator system. It was found that, at the inclusion of five or more modules, with a 170 ms delay, the fifth module was still acting on a previous iteration of data as the first module received new data. This limitation, although able to be designed out with faster microprocessing, put a practical limit on the number of modules to be used in the proof of concept evaluation.

6 Electronics

The robotic arm's electronics was designed to parallel the modular framework of the manipulator. The embedded module electronics were set up in a serial network, where, firstly, data passed from the camera PCs to the top module and then, sequentially via SPIM/SPIS, down the manipulator. Power was also parallel daisy-chained between modules with smoothing capacitors and local voltage regulation keeping power clean of noise.

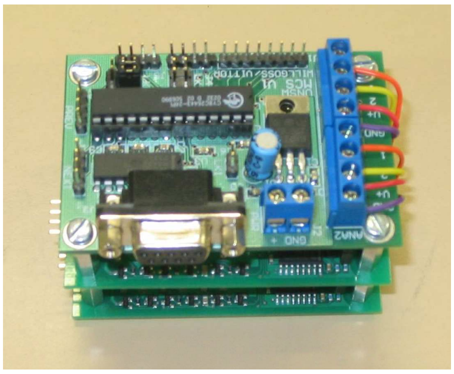

Figure 8. Three Board Electronic Module Stack

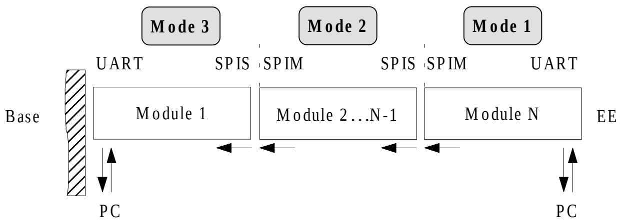

Figure 9. Module Wiring Mode Placements

Although the original design criterion was to contain all parts of the module inside the cylindrical covers, this was not possible for the proof of concept stage without large financial and development expenditure. The smallest standard control boards for each 3-phase motor plus an in-house designed microprocesser board were stacked and attached to the side of the modules. The individual module electronics were assembled in stacks as shown in Figure 8, being 32mm high, 51mm wide and 74mm long.

The communication network was set up to contain three optional module configurations. Figure 9 illustrates the placement of these three module configurations within a manipulator system. The system's wiring allowed PC communication at each end of the serial multi-module processor network as configured for this testing regime. Future concept testing could involve building four to six serial manipulators as shown here and linking them as for say a Stewert platform or limbs to a quadruped. The same clone modules could be used by simply changing a few jumpers on each microprocessor board.

The top board in each stack contained the microprocessor and main controller interfaces for each module. The microprocessor on each module was used to read inputs from the module's sensors, communication between adjacent modules, motor control and communication with end effector input/output.

7 Actuation

Two primary design criteria have been to generate compact and robust module motion. The actuation loop, as shown in Figure 10, shows the components required to provide actuation to the two DOF module joints.

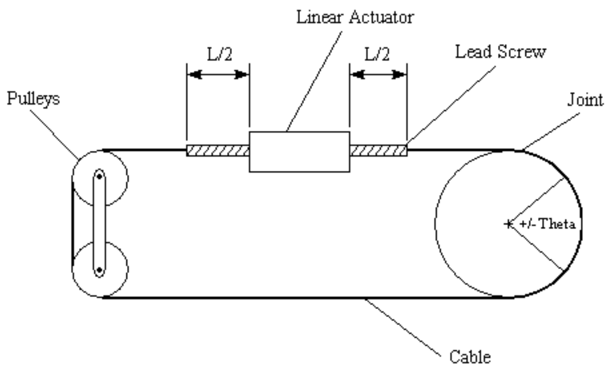

Figure 10. Ball Screw and Steel Cable Interface

The actuation loop consisted of two sections. The first comprised a closed loop position control with optical encoder feedback. This loop began with the microprocessor determining error and required velocity. A three-phase brushless motor; the required commutative signals generated by the respective motor controller, was used to drive a planetary gear head, the angular displacement being measured using an optical encoder. Pulses from the optical encoders were fed to the microprocessor and counted, noting the direction of actuation taking place. The second section of actuation was an open loop system. The respective planetary gear heads were coupled to linear ball screws. Linear motion provided by the pretensioned steel cables was converted into rotary motion in the 2 DOF joint. Three aspects of the effectiveness of the actuation loop were considered in this design.

Firstly, the accuracy of the feedback through the optical encoder relative to the motion of the joint was investigated. It was found that the motion of the joint could be controlled to 0.01 degrees. This was deemed suitable for the functional requirements of the module. Further, the software counter could only be controlled to an accuracy of +/-100 counts. Therefore the joint was necessarily controlled to an accuracy of +/-1 degree.

Secondly, the strength capabilities of each module were addressed to determine the strength/weight required of each module and the maximum possible number of modules that could be attached serially. Four possible manipulator configurations were explored, shown in Figure 11, allowing varying strength requirements on each module to be considered.

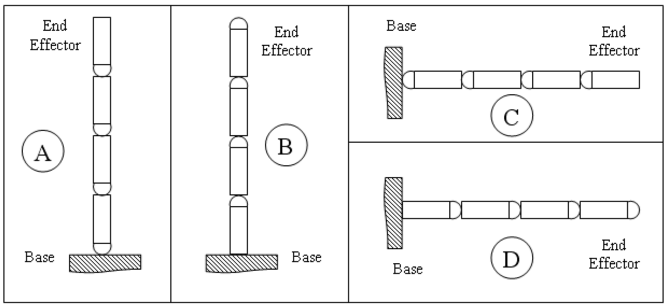

Figure 11. Possible Manipulator Configurations

For the four possible manipulator configurations, static loading of the joints was calculated for maximum joint angles of +/-20 degrees. The maximum torque requirement occurred in case C (see Figure 11). Pretensioning of the cables by repositioning of the pulleys ensured there was never a point during the motion of the module when the cable went slack. Modules were observed to reverse motion direction without hysteresis or impulses being introduced into the system.

Thirdly, the hysteresis within the actuation system was investigated. Three main causes of hysteresis and their eventual mitigations are listed below.

- Undesired rotation of the ball screw - linear guides introduced on the steel cable.
- Slack in joint ball bearings under axial load - angle bearings used.
- Play between the swaged ends on the ball screw – improvement of manufacturing tolerances.

Notwithstanding all the design criteria investigated above, it was discovered that the ultimate limiting factor was the fine control of the motors. The control at high speed required, within the confines of the compact controller board, a turning on/off of power to the motor for motion reversal. This became a serious inhibiting factor for responsive control because it took 20mS for each motor to lock on to phases after being turned on again which itself introduced a short jerky motion to the actuator. Full bipolar control with the motors running continuously was possible but by using control boards dimensionally five times the size of those used here. To use the full dynamic range of the motor, high ratio gearboxes were fitted that overcame some of the motor start-up problems and provided more mechanical advantage to movement. However, maximum speed was reduced but the movement was considered, as before, to represent a kinematic quasi-static path.

The desired joint motion profiles were calculated for a five-module manipulator in the worst case manipulator configuration. It was found that the maximum velocity in the motion profile of each module joint was limited to 20 degrees.s^{-1} and the maximum acceleration of each joint was 50 degrees.s^{-2}. This meant that the maximum motion range of 40 degrees could be covered in a minimum period of 2.4s.

8 Results

To be representative of hyper-redundancy and stay within mechanical load bearing limitations, a maximum of four modules were tested in the manipulator system. Figure 12 shows the assembled four-module manipulator system in its home configuration, i.e. all angles set to zero. The top module shows a stereo vision system mounted upon it. The following three tests were undertaken with the manipulator:

- Target tracking
- Obstacle Avoidance
- Fault tolerance

For the first case the manipulator system was set up such that each module in the system tracked the goal position using MDC. When the goal position was moved, the manipulator successfully tracked the goal point in 3D space. Some compliance was still present in the system. However the stereo vision system accommodated for these errors, with the end effecter successfully settling on the goal position [see Video 1]. The two scenarios demonstrated in the manipulator's target tracking included:

- Point to point end effecter motion.
- End effecter trajectory following.

For the second case the manipulator system was again set up such that each module in the system tracked the goal position using MDC. However, during goal tracking, individual module motion was interrupted through pressing their respective local proximity sensors. Motion successfully accommodated these external interrupts, modifying the global manipulator configuration while continuously moving the end effecter to the goal. Obstacle avoidance in the MDC approach of a manipulator system was achieved both for static and dynamic end effecter goal positioning [see Videos 2a and 2b].

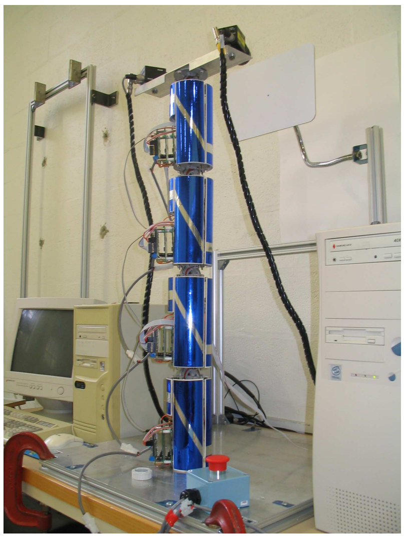

Figure 12. Four Module Assembled Manipulator System

For the third case, the system was modified to simulate hardware failure in the manipulator. The first and second cases were then successfully repeated to demonstrate MDC in slow non-catastrophic system degradation. The video shows movement where one module does not take any corrective action. [see Video 3].

9 Conclusions

This paper has outlined the successful design, manufacture and implementation of a reconfigurable modular manipulator system, normally termed a proof of concept, using MDC as the control scnario. The clone hardware, actuation, software and electronic sub-systems were, for the most part, successfully integrated into one manipulator system. The implemented methodology demonstrated goal target tracking, obstacle avoidance and fault tolerance.

10 Future Work

There are many aspects of the project that could be improved once proof of concept is achieved. If the end effector sensor remains a stereo vision input, miniaturisation and reduced image processing time need to be achieved to increase speed of motion. An increase in speed of manipulator motion is also possible by using motors with improved control. The present motors have limitations in maitaining continuous motion at very low speed, all of which has stemmed from keeping power to weight and size within the present module specification. This phase may be an expensive and time-consuming exercise.

The mechanical design could be made to increase the range or motion for each joint and reduce compliance in system hardware. To improve hardware performance, the system would also benefit from a non-contact proximity sensing. To improve assembly, disassembly and reconfigurability of the manipulator system, simplified plug and play interfaces between the modules and their casings would be needed but will be addressed in terms of a full commercialisation of the concept.

Once module design can include some of the improvements noted above, it will be possible to assemble systems with parallel manipulator structures such as the Stewart Platform or multipeds. Finally the issue of scaling the size and capabilities of the modules in order to maximise the length of the manipulator and its weight carrying capacity for a specific application needs to be undertaken.

We are now liaising with venture capitalists and international robotics companies to build a working prototype i.e. to demonstrate the advantages of MDC in a full industrial setting.

Videos are at URL www.youtube.com
account acra20007; named MDC Robot clip 1,2a,2b and 3.

References

[Chen, & Yang, 1996] I.M, Chen. & G, Yang. "Configuration Independant Kinematics for Modular Robots" , *IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota*. 1996

[Chen, et el., 1998] I.M, Chen. C. Y., Chen. G, Yang. W, Chen. I.-G, Kang. S. H, Yeo. and G, Chen. "Architecture for Rapidly Reconfigurable Robot Workcell," *5th Int. Conf. Control, Automation, Robotics and Vision, Singapore*, pages 100-104. 1998

[Chen, 2000] I.-M, Chen. "Realization of a Rapidly Reconfigurable Robotic Workcell," *Journal of Japan Society for Precision Engineering*, Vol.66, No.7, pages1024-1030. 2000.

[Hayashi, et al, 2002] S, Hayashi. K, Ganno. Y, Ishii. & I, Tanaka. 2002, "Robotic Harvesting System for Eggplants" JARQ , Volume 36, No. 3, pages 163-168.

[Henten et al., 2002] Eldert J. Van Henten, Bart A.J. Van Tuijl, Jochen Hemming, Jan G. Kornet, and Jan Bontsema, "CUPID – An Autonomous Cucumber Picking Device" *Proceedings of Mechatronics, University of Twente*, June, 2002.

[Kelmar and Khosia, 1988] L. Kelmar, and P. Khosia, "Automatic generation of kinematics for areconfigurable modular manipulator system" *Proc. IEEE Int. Conf. Robotics and Automation, hiladelphia*, USA, 1988.

[Kelmar and Khosia, 1990] L. Kelmar, and P. Khosia, "Automatic generation of forward and inverse kinematics for a reconfigurable modular manipulator system" *Journal of Robotics Systems* Vol.7 No.4, pp. 599-619, 1990.

[Marzwell and Slifko, 1995] N.I. Marzwell, and A. Slifko, "Visual Inspection Planar Serpentine Manipulator" *NASA's Technology 2005 conference*, Chicago, IL., October, 1995.

[Matsuno and Suenaga, 2003] F. Matsuno and K. Suenaga, "Control of Redundant 3D Snake Robot based on Kinematic Model" *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September, 2003.

[Mclean and Cameron, 2003] A. Mclean, and S. Cameron, "Path Planning and Collision Avoidance for Redundant Manipulators in 3D" *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September, 2003.

[Mavroidas, et al., 1995] C.Mavroidas, P.Rowe, and S.Dubowsky, "Inferred end-point control of long reach manipulators," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems Pittsburgh*, PA, pp. 71-76, Aug. 1995.

[Paredis et al., 1996] C.J.J. Paredis, H.B. Brown and P.K. Khosla, "A Rapidly Deployable Manipulator System" *IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota*, April, 1996.

[Sarig, 1993] Y. Sarig, "Robotics of Fruit Harvesting: A State-of-the-art Review" *Journal of Agricultural Engineering Research*, 54, pp. 265-280, 1993.

[Schmitz, et al., 1989] D. Schmitz, P. Khosia, R. Hoffman, and T. Kanade, "CHIMERA: A real-time programming environment for manipulator control", *Proc. IEEE Int. Conf. Robotics and Automation*, Vol.2, pp. 846-852, 1989.

[Vittor, et al., 2003] T.Vittor, R.Willgoss and T.Furukawa, "Modular Decentralized Control of Fruit Picking Redundant Manipulator" *Australian Conference on Robotics and Automation*, Brisbane, December, 2003.

[Vittor, et al., 2004] T.Vittor, R.Willgoss and T.Furukawa, "Hyper-Redundant Manipulator Control

For Reconfigurability and Obstacle Avoidance" *Australian Conference on Robotics and Automation*, Canberra, December, 2004.

[Vittor and Willgoss, 2005] T.Vittor and R.Willgoss, "Stability and Motion Analysis for Decentralized Control of Modularized Manipulators", *Int. Conf. Computational Intelligence, Robotics and Autonomous Systems*, Singapore, December, 2005.

[Vittor, & Willgoss, 2005b] T.Vittor and R.Willgoss, "Motion Analysis for Decentralized Control of N-Module Hyper-Redundant Manipulators" *Australian Conference on Robotics and Automation*, Sydney. 2005

[William II and Mayhew IV, 1997] R.L. William II, and J.B. Mayhew IV, "Obstacle-Free Control Of The Hyper-Redundant Nasa Inspection Manipulator" *Proc. of the Fifth National Conf. on Applied Mechanics and Robotics*, October 12-15, 1997.

[Yamakita, et al., 2003] M. Yamakita, M. Hashimoto and T. Yamada., "Control of Locomotion and Head Configuration of 3D Snake Robot (SMA)" *IEEE International Conference on Robotics and Automation*, September, 2003.

[Yook, et al., 1997] J. Yook, D. Tilbury, K. Chervela and N. Soparkar, "Decentralized, Modular Real-Time Control for Machining Applications" *Proceedings of the American Control Conference* pp 844-849. 1997.