2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Modular Robot Manipulators Based on Virtual Decomposition Control

Wen-Hong Zhu, *Senior Member, IEEE*, and Tom Lamarche

***Abstract*— Modular or re-configurable robots have been studied and developed over two decades. Most researches focus on mechatronic interfaces and re-configurable capabilities. However, less attention has been paid to dynamics and control. Consequently, the control performance of a modular robot has never been comparable with an integrated robot, due to the lack of proper handling of the dynamic interactions among the modules. In this paper, the application of the *virtual decomposition control* to modular robot manipulators is discussed. A high-speed databus with a data rate of 100 Mbps is used for necessary information exchange among the modules. The dynamics based control is fully handled by the local embedded controllers, whereas the host computer handles the kinematics related computation. The stability of the entire robot is rigorously guaranteed. This research aims at giving the modular robots the comparable control performance as the integrated robots, while keeping the fundamental feasibilities such as low cost for mass production, high flexibility, and easy use and expansion.**

I. INTRODUCTION

Industrial robot manipulators are generally called integrated manipulators, since not only their mechanical structure, sensors and actuators, but also the electronics and computing systems are pre-designed and fabricated before delivered to users. While possessing relatively high precision in operation, the integrated robot manipulators are vulnerable to component/part failures, possess low flexibility against task-variation and are unable to adapt structure changes required by on-site applications. Whereas these drawbacks do not cause problems for robot-oriented manufacturing lines where robots are programmed to perform repeated tasks, they do limit the adaptability, flexibility, and versatility of the robots when operated in unstructured environments.

Most commercially available robot manipulators are generally stand-along devices with separate control cabinets. This aspect makes it difficult to incorporate them onto mobile platforms. The fact of having fixed structure makes it difficult to perform a variety of tasks that require different robot structures with different motion degrees of freedom. When a component, such as a sensor or an actuator, of an integrated manipulator goes out of order, the solution is either to repair the component or to replace the entire manipulator. Repairing might be a difficult task in a hostile environment (such as in a space environment) where human access is highly restricted.

A possible solution is to use modular robot manipulators [1]-[6]. A modular robot manipulator simply consists of robot modules. These modules incorporate actuators, sensors, and electronics into their mechanical assemblies to create all-in-one components which can be appropriately connected to form any desired robot manipulator. Fukuda $et.\ al.$ introduced the concept of cellular robotics in [1]. Paredis and Khosla developed a reconfigurable modular manipulator system (RMMS) with modular links and joints [2]. Kinematic configuration optimization was performed analytically for robots with two degrees of freedom (DOF) and performed numerically for robots with multiple DOF. In the meantime, a (kinematic) configuration optimization procedure based on an *assembly incidence matrix* representation was presented by Chen and Burdick [3]. A genetic algorithm leading to task-based design was proposed by Chung $et.\ al.$ [4]. Yim $et.\ al.$ developed a reconfigurable modular robot PolyBot targeting space applications [5]. Recently, a modular joint design for a humanoid robot was reported by Lohmeier $et.\ al.$ [6].

Modular robot manipulators will have at least four features over conventional integrated robot manipulators:

i) the integration of electronics into mechanical assembly makes the modular robot manipulators particularly suitable for mobile applications. Without requiring additional space for its electronics and controller, a modular robot manipulator can be conveniently mounted on any mobile platform as long as the power and data connectors are accessible.
ii) modular robot manipulators allow on-site change of robot structure and even change of motion degrees of freedom by adding/removing modules to/from the robots.
iii) it makes easy repair possible, since a defected module can be easily replaced.
iv) mass production of modules will eventually reduce the cost of using modular robot manipulators.

Besides for ground applications, above features make the modular robot manipulators particularly suitable for space applications. In space, due to the limited accessibility and the substantial cost for human presence, use of robots becomes more feasible. In the future, it is likely that the robotic technologies will be used not only to acquire information, but also to manipulate space/planetary environments. Thus, adaptability, flexibility, reliability, and re-formability will be the fundamental requirements. Therefore, in future space robotic missions, mission planners might elect to send robot modules instead of integrated robots themselves into space, and to assemble these modules in space to form any type of robots needed for operations, with minimum human intervention.

The authors are with the Spacecraft Engineering, Space Technologies, Canadian Space Agency, 6767 route de l'Aeroport, Saint-Hubert, QC J3Y 8Y9, Canada (e-mail: Wen-Hong.Zhu@space.gc.ca, Tom.Lamarche@space.gc.ca)

After its concept being initialized over two decades ago, most existing modular robots focus mainly on kinematic functions, such as the reconfigurability and reformability, and the kinematic configuration optimization. The modularity is mainly reflected by the aspect of electro-mechanical design. Each module has very limited capability of local communications to its neighbors, and is usually implemented with a very simple local controller (such as a proportional-integral-derivative (PID) controller). While the communication and control issues are less sensitive to certain applications such as snake robot climbing, they are substantially important for applications where precise position control and multi-module coordinated control are necessarily required. Although some commercial products like PowerCube[1] are available, their control performance is far from reaching the accuracy delivered by dynamics-based control.

The *virtual decomposition control* [7], [8] (VDC) is an approach that uses subsystem dynamics to conduct control while rigorously guaranteeing the L_2 and L_∞ stability of the entire system. In this paper, the VDC approach is applied to handle the dynamics-based control issue of modular robot manipulators. This application allows dynamics-based control to be implemented in local modules with the support of a high-speed communication system that efficiently transfers kinematics/dynamics data among the modules. When these modules are integrated to form a robot manipulator, the master computer only computes the robot kinematics including velocity and force transformations among the modules. Despite all the module dynamics are handled locally within the modules themselves, the stability of the entire robot system is rigorously guaranteed to deliver dynamics-based control performance.

Finally, it has to be acknowledged that the communication system should have a reasonably high data rate in order to keep a high sampling rate for control. The currently available industrial standard buses like the Controller Area Network[2] (CAN) and the SErial Realtime COmmunication System[3] (SERCOS) have data rates about 1-16 Mbps, which is not sufficient enough to be used. Based on a real time test with 12 modules, the communication data rate has to be at least 100 Mbps. The field programmable gate array[4] (FPGA) based Q5 board from Xiphos Technologies Inc. satisfies this requirement.

This paper is organized as follows. In Section II, preliminary basis in [7] is summarized. A typical robot module comprised of two rigid links and one joint together with associated electronics and sensor/actuator is described in Section III. In Section IV, the application of the VDC approach to modular robot manipulators is outlined in technical details with stability analysis. A system test with two modules supported by a SpaceWire[5] based high-speed communication system is presented in Section V, which enables the VDC based control to be effectively implemented.

[1]See: http://www.amtec-robotics.com/index.html
[2]See: http://www.semiconductors.bosch.de/en/20/can/index.asp
[3]See: http://www.sercos.de/english/index.htm
[4]See: http://www.xilinx.com
[5]See: http://spacewire.esa.int/tech/spacewire

II. PRELIMINARY BASIS

Contents in this section are cited from [7].

A. Body Frame Expression

Let

$$^AF = [^Af^T, ^Am^T]^T = [(^AR_I f_A)^T, (^AR_I m_A)^T]^T \in \Re^6$$

be a generalized force/moment measured and expressed in frame $\{A\}$, where $^AR_I \in \Re^{3\times3}$ denotes a rotation transformation matrix which transforms a 3×1 vector expressed in the inertial frame I to that expressed in frame $\{A\}$. In the meantime, this generalized force/moment can be measured and expressed in another frame called frame $\{B\}$ through a transformation matrix ^BU_A as

$$^BF = {}^BU_A{}^AF \tag{1}$$

where

$$^BU_A = \begin{bmatrix} ^BR_A & 0 \\ (^Br_A\times)^BR_A & ^BR_A \end{bmatrix} \in \Re^{6\times6}$$

and $^Br_A \in \Re^3$ denotes a vector from the origin of frame $\{B\}$ to the origin of frame $\{A\}$, expressed in frame $\{B\}$.

The generalized linear/angular velocities of frame $\{B\}$ and expressed in frame $\{B\}$ is defined as

$$^BV = [^Bv^T, ^B\omega^T]^T = [(^BR_I v_B)^T, (^BR_I\omega_B)^T]^T \in \Re^6.$$

If frames $\{A\}$ and $\{B\}$ are fixed to a common rigid body, it follows that

$$^AV = {}^BU_A^T{}^BV. \tag{2}$$

B. Rigid Body Dynamics in Body Frame

The net force/moment of a rigid body expressed in a body-fixed frame $\{A\}$ can be written as

$$^AF^* = M_A\frac{d}{dt}(^AV) + C_A(^A\omega)^AV + G_A \tag{3}$$

where the detail expressions of M_A, $C_A(^A\omega)$, and G_A have been given in [7].

Let $^AV_r \in \Re^6$ be the required vector of $^AV \in \Re^6$. Define

$$Y_A\theta_A \stackrel{def}{=} M_A\frac{d}{dt}(^AV_r) + C_A(^A\omega)^AV_r + G_A \tag{4}$$

where $Y_A \in \Re^{6\times13}$ is a regressor matrix and $\theta_A \in \Re^{13}$ is a vector containing all dynamic parameters (with three dependent parameters) of the rigid body.

C. Virtual Power Flow

Definition 1: The *virtual power flow* (VPF) at frame $\{A\}$ is defined as the inner product between the generalized velocity error and the generalized force error, that is

$$p_A = (^AV_r - {}^AV)^T(^AF_r - {}^AF) \tag{5}$$

where the subscript r represents the corresponding required (design) vector.

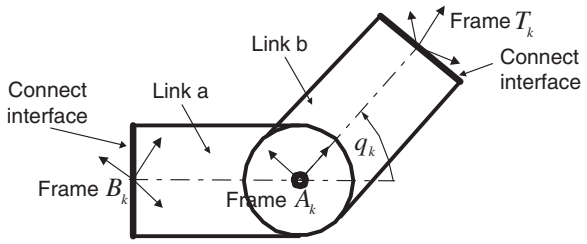

Fig. 1. A module assembly.

III. MODULE FUNCTIONS

A robot manipulator consists of a variety of individual modules connected serially. A typical module is illustrated in Fig. 1. It is comprised of two rigid links connected by a joint. A standard electro-mechanical interface is built at each of the two ends. These interfaces mechanically fix a module with another while providing high-speed communication and electrical power connections between the two modules. The mechanical connection uses screw-bolt type to maintain good link rigidity. High-speed communication goes through a standard D-Sub 9 pins connector (same as used for PC serial ports) while the other connector provides 24V DC power connection, see Fig. 2. Each module's electronics comprises a micro-computer board to handle communication and control, local DC/DC converters to generate isolated logic-level voltage supplies, and a power amplifier to drive the joint's DC brushless motor, see Fig. 3. Each joint's motor-transmission assembly has a position encoder attached to the motor side. Once all the modules are connected together, the overall electrical/electronic interface is simply two connectors: DC power (2 pins) and high-speed databus (9 pins). This feature substantially increases the mobility of the robot since no control cabinet is required (only a 24V power supply) and wiring is minimal.

Three coordinate frames are attached to each module. Frames $\{B_k\}$ and $\{T_k\}$ are attached to the two connection interfaces of the kth module, respectively. Frame $\{A_k\}$ is attached to the same link frame $\{T_k\}$ is attached, with its z axis aligning with the joint axis.

The complete modular robot manipulator is comprised of n modules. The interface of frame $\{B_1\}$ is attached to the base; the interface of frame $\{T_1\}$ is connected with the interface of frame $\{B_2\}$; the interface of frame $\{T_{k-1}\}$ is connected with the interface of frame $\{B_k\}$; and eventually the interface of frame $\{T_{n-1}\}$ is connected with the interface of frame $\{B_n\}$.

IV. COMMUNICATION AND CONTROL

The overall communication diagram is illustrated in Fig. 4. Each slave node represents the micro-computer-board of a module. The sole master node is a real-time computer connected to the first slave node of the robot through a high-speed databus.

The communication protocol is designed in such a way that the master node is able to broadcast information to all

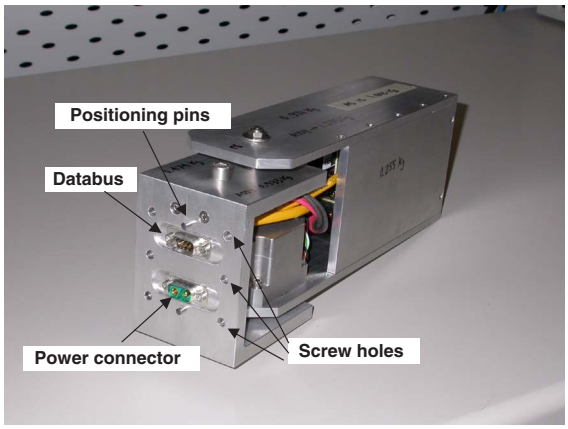

Fig. 2. Module interface.

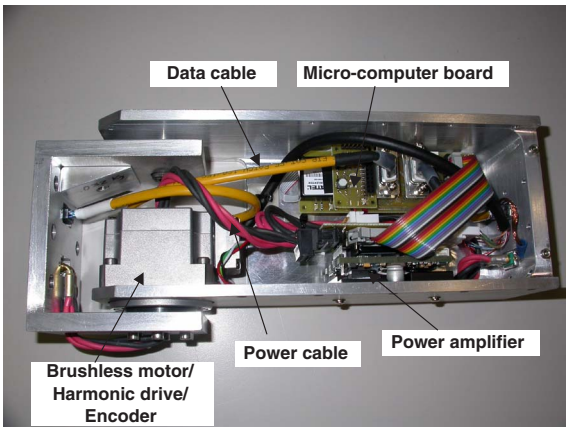

Fig. 3. Module internal view.

the slave nodes, and is able to extract information from any designated slave node.

With respect to this communication protocol, two cycles with four actions within each sampling period are designed in order to have the *virtual decomposition control* implemented. The first communication cycle constitutes *Communications A* and *B*, and the second communication cycle constitutes *Communications C* and *D*.

A. Communication and Control Computations

The complete communication and control computation loop within a sampling period starts by executing the first communication cycle. The master node extracts joint positions and velocities from all the slave nodes through *Commu-*

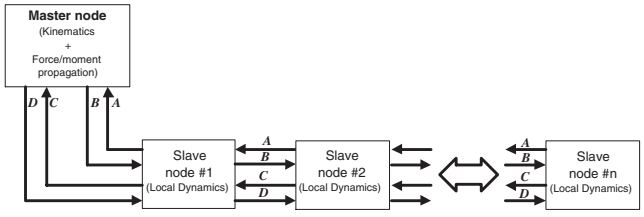

Fig. 4. The communication system.

nication A, then the master node computes the linear/angular velocities and the corresponding required velocities and accelerations in body frames of all the slave nodes, and broadcasts them back to all the slave nodes through *Communication B*. After receiving its own body-frame-based velocity, required velocity, and required acceleration, each slave node computes its required net force/moment based on the local dynamics of each module. The second communication cycle starts by extracting the net forces/moments of all the slave nodes into the master node through *Communication C*. Then the master node computes the required projection torques of all the joints, and sends them back to each slave node through *Communication D*. Finally, each slave node computes its control torque based on the dynamics of each individual joint and by using the required projection torque.

The detailed procedure is listed below:

Step 1: Perform *Communication A*, through which the master node extracts the joint positions $q_k \in \Re$ and joint velocities $\dot{q}_k \in \Re$, $k \in \{1, n\}$ from all the slave nodes.

Step 2: In the master node, for given $q_{kd} \in \Re \cap L_\infty$, $\dot{q}_{kd} \in \Re \cap L_\infty$, and $\ddot{q}_{kd} \in \Re \cap L_\infty$, design

$$\dot{q}_{kr} = \dot{q}_{kd} + \lambda_k (q_{kd} - q_k) \tag{6}$$

for $k \in \{1, n\}$, with $\lambda_k > 0$.

From (2), the velocity transformation from frame $\{B_k\}$ to frame $\{B_{(k+1)}\}$ can be written as[6]

$${}^{B_{(k+1)}}V = {}^{A_k}U^T_{B_{k+1}} z \dot{q}_k + {}^{B_k}U^T_{B_{k+1}} {}^{B_k}V \tag{7}$$

with $z = [0, 0, 0, 0, 0, 1]^T \in \Re^6$.

Accordingly, the required velocity transformation from frame $\{B_k\}$ to frame $\{B_{(k+1)}\}$ are written as

$${}^{B_{(k+1)}}V_r = {}^{A_k}U^T_{B_{k+1}} z \dot{q}_{kr} + {}^{B_k}U^T_{B_{k+1}} {}^{B_k}V_r. \tag{8}$$

For given ${}^{B_1}V = 0$ and ${}^{B_1}V_r = 0$, computing (7) and (8) recursively from $k = 1$ to $k = n-1$ yields ${}^{B_k}V \in \Re^6$ and ${}^{B_k}V_r \in \Re^6$, $k = 2, 3, \cdots, n$.

Furthermore, differentiating (6) and (8) with respect to time yields $\frac{d}{dt}\left({}^{B_k}V_r\right) \in \Re^6$, $k = 2, 3, \cdots, n$, from given $\frac{d}{dt}\left({}^{B_1}V_r\right) = 0$.

Step 3: Perform *Communication B*. Transfer ${}^{B_k}V \in \Re^6$, ${}^{B_k}V_r \in \Re^6$, $\dot{q}_{kr} \in \Re$, $\frac{d}{dt}\left({}^{B_k}V_r\right) \in \Re^6$, and $\ddot{q}_{kr} \in \Re$ to the kth slave node for $k \in \{1, n\}$.

Step 4: In the kth slave node, $k \in \{1, n\}$, the following computations are performed first

$$\begin{aligned} {}^{T_k}V &= {}^{A_k}U^T_{T_k} z \dot{q}_k + {}^{B_k}U^T_{T_k} {}^{B_k}V \quad (9) \\ {}^{T_k}V_r &= {}^{A_k}U^T_{T_k} z \dot{q}_{kr} + {}^{B_k}U^T_{T_k} {}^{B_k}V_r \quad (10) \end{aligned}$$

from given ${}^{B_k}V \in \Re^6$ and ${}^{B_k}V_r \in \Re^6$.

Furthermore, differentiating (10) with respect to time yields $\frac{d}{dt}\left({}^{T_k}V_r\right) \in \Re^6$ from given $\frac{d}{dt}\left({}^{B_k}V_r\right) \in \Re^6$.

[6]Note that once the kth slave node is connected with the $(k+1)$th slave node, frames $\{B_{(k+1)}\}$ and $\{T_k\}$ are viewed as fixed to the same rigid body by having constant ${}^{T_k}U_{B_{k+1}} \in \Re^{6\times 6}$.

Compute the required net force/moments for the two rigid links

$$\begin{aligned} {}^{B_k}F^*_r &= Y_{B_k}\hat{\theta}_{B_k} + K_{B_k}\left({}^{B_k}V_r - {}^{B_k}V\right) \quad (11) \\ {}^{T_k}F^*_r &= Y_{T_k}\hat{\theta}_{T_k} + K_{T_k}\left({}^{T_k}V_r - {}^{T_k}V\right) \quad (12) \end{aligned}$$

where $K_{B_k} \in \Re^{6\times 6}$ and $K_{T_k} \in \Re^{6\times 6}$ are two positive-definite gain matrices characterizing velocity feedback control; $Y_{B_k}\hat{\theta}_{B_k}$ and $Y_{T_k}\hat{\theta}_{T_k}$ denote the model based feedforward compensations defined by (4) with appropriate frame substitution.

Define

$$\begin{aligned} s_{B_k} &= Y^T_{B_k}\left({}^{B_k}V_r - {}^{B_k}V\right) \quad (13) \\ s_{T_k} &= Y^T_{T_k}\left({}^{T_k}V_r - {}^{T_k}V\right). \quad (14) \end{aligned}$$

The $\mathcal{P}$ function defined in [8] is used to update the elements of $\hat{\theta}_{B_k} \in \Re^{13}$ and $\hat{\theta}_{T_k} \in \Re^{13}$ as:

$$\hat{\theta}_{B_k\gamma} = \mathcal{P}\left(s_{B_k\gamma}, \rho_{B_k\gamma}, \underline{\theta}_{B_k\gamma}, \overline{\theta}_{B_k\gamma}\right), \ \gamma \in \{1, 13\} \tag{15}$$

$$\hat{\theta}_{T_k\gamma} = \mathcal{P}\left(s_{T_k\gamma}, \rho_{T_k\gamma}, \underline{\theta}_{T_k\gamma}, \overline{\theta}_{T_k\gamma}\right), \ \gamma \in \{1, 13\} \tag{16}$$

where $\hat{\theta}_{B_k\gamma}$ denotes the γth element of $\hat{\theta}_{B_k}$ and $\hat{\theta}_{T_k\gamma}$ denotes the γth element of $\hat{\theta}_{T_k}$; $s_{B_k\gamma}$ denotes the γth element of s_{B_k} defined in (13) and $s_{T_k\gamma}$ denotes the γth element of s_{T_k} defined in (14); $\rho_{B_i\gamma} > 0$ and $\rho_{B_i\gamma} > 0$ are update gains; $\underline{\theta}_{B_k\gamma}$ and $\overline{\theta}_{B_k\gamma}$ denote the lower and upper bounds of $\theta_{B_k\gamma}$ - the γth element of $\theta_{B_k} \in \Re^{13}$; and $\underline{\theta}_{T_k\gamma}$ and $\overline{\theta}_{T_k\gamma}$ denote the lower and upper bounds of $\theta_{T_k\gamma}$ - the γth element of $\theta_{T_k} \in \Re^{13}$.

Define and compute the following new vector

$$F_{kr} = {}^{B_k}F^*_r + {}^{B_k}U_{T_k} {}^{T_k}F^*_r. \tag{17}$$

Step 5: Perform *Communication C*, through which the master node extracts $F_{kr} \in \Re^6$, $k \in \{1, n\}$, from all the slave nodes.

Step 6: In the master node. compute the following force/moment propagation

$$z^T {}^{B_k}F_r = z^T \sum_{j=k}^{n} {}^{B_k}U_{B_j} F_{jr} \tag{18}$$

from $k = n$ to $k = 1$ to obtain the required projection torques for all the joints.

Step 7: Perform *Communication D*, through which the required projection torques for all the joints, denoted as $z^T {}^{B_k}F_r \in \Re$, $k \in \{1, n\}$, are sent to all the slave nodes.

Step 8: In the kth slave node, $k \in \{1, n\}$, compute the joint control torque

$$\tau_k = Y_k\hat{\theta}_k + z^T {}^{B_k}F_r + k_k(\dot{q}_{kr} - \dot{q}_k) \tag{19}$$

with $k_k > 0$. In (19), $\hat{\theta}_k \in \Re^4$ denotes the estimates of $\theta_k \in \Re^4$ expressed as

$$\begin{aligned} Y_k &= \begin{bmatrix} \ddot{q}_{kr} & \text{sign}(\dot{q}_{kr}) & \dot{q}_{kr} & 1 \end{bmatrix} \quad (20) \\ \theta_k &= \begin{bmatrix} I_{mk} & k_{ck} & k_{vk} & c_k \end{bmatrix}^T \quad (21) \end{aligned}$$

where $I_{mk} \in \Re$ is the equivalent rotation inertia; $k_{ck} > 0$ and $k_{vk} > 0$ denote the Coulomb and viscous friction coefficients, respectively, and $c_k \in \Re$ denotes a DC offset that accommodates asymmetric Coulomb friction.
Define

$$s_k = Y_k^T \left(\dot{q}_{kr} - \dot{q}_k \right). \tag{22}$$

The $\mathcal{P}$ function defined in [8] is used to update the 4 parameters of $\hat{\theta}_k \in \Re^4$ as follows:

$$\hat{\theta}_{k\gamma} = \mathcal{P}\left(s_{k\gamma}, \rho_{k\gamma}, \underline{\theta}_{k\gamma}, \overline{\theta}_{k\gamma} \right), \ \gamma \in \{1, 4\} \tag{23}$$

where $\hat{\theta}_{k\gamma}$ denotes the γth element of $\hat{\theta}_k$ and $s_{k\gamma}$ denotes the γth element of s_k defined in (22); $\rho_{k\gamma} > 0$ is an update gain; $\underline{\theta}_{k\gamma}$ and $\overline{\theta}_{k\gamma}$ denote the lower and upper bounds of $\theta_{k\gamma}$ - the γth element of $\theta_k \in \Re^4$.

Besides the four communication actions in Steps 1, 3 ,5, and 7, computations in the master node are mainly reflected in Steps 2 and 6, and computations in each slave node are reflected in Steps 4 and 8 where parallel computations are permitted due to the FPGA based implementation.

Note that Steps 2 and 6 involve kinematics computations only, where the force/moment transformation matrices are used. All the dynamics based computations are reflected in Steps 4 and 8. This is the main feature of using the *virtual decomposition control* in modular robot applications. All the dynamics are handled by local modules. The master node handles the kinematics related computations only.

B. Practical Implementation

The complete communication and control computation has to be completed within a sampling period. The size of the communication package is very critical to meet the timing constraints. Therefore, the information exchange in the communication actions should be kept as minimal as possible. In practical implementation, $\dot{q}_k$ is removed from *Communication A* and will be re-created numerically in Step 2. Also, $\frac{d}{dt}\left({}^{B_k}V_r\right)$ and $\ddot{q}_{kr}$ are removed from *Communication B* and will be re-created numerically in Step 4.

After the modification, it can be verified that the numbers of variables that need to be transferred between the master node and each of the slave nodes in the four communication actions are 1, 13, 6, and 1, respectively. With a 24-bit resolution for the joint position and a 16-bit resolution for all other variables, a 26-byte data length is required. To make the communication package uniform and to provide a little room for possible expansion, a 40-byte package is designed for all communication actions between the master node and each slave node. If the communication time spent on data transfer for a 12-module robot is limited to 200 micro-seconds, it requires the databus speed to be at least

$$\begin{aligned} & \frac{8 \text{ bits}}{\text{byte}} \times \frac{40 \text{ bytes}}{\text{package}} \times \frac{4 \text{ packages}}{\text{node}} \times 12 \text{ nodes} \times \frac{1}{200\mu s} \\ = \ & \frac{8 \times 40 \times 4 \times 12}{200 \times 10^{-6}} \\ = \ & 76.8 \text{ (MHz)}. \end{aligned}$$

C. Stability

The stability results are summarized as a Proposition:

Proposition 1: Consider a modular robot manipulator comprised of n modules described in Section III, subjected to the VDC-based communication and control outlined from Step 1 to Step 8 in Subsection IV-A. It follows that

$$q_{kd} - q_k \in L_2 \cap L_\infty \tag{24}$$
$$\dot{q}_{kd} - \dot{q}_k \in L_2 \cap L_\infty \tag{25}$$
$$q_{kd} - q_k \to 0 \tag{26}$$
$$\dot{q}_{kd} - \dot{q}_k \to 0 \tag{27}$$

for $\forall k \in \{1, n\}$. ■

The proof is described briefly as follows.

A non-negative accompanying function

$$\nu_{mk} = \nu_{B_k} + \nu_{T_k} + \nu_k \tag{28}$$

is assigned to the kth slave node, $k \in \{1, n\}$, where $\nu_{B_k} \geq 0$ and $\nu_{T_k} \geq 0$ are assigned to the two links, respectively, and $\nu_k \geq 0$ is assigned to the joint. The three non-negative function have expressions as

$$\begin{aligned} \nu_{B_k} = & \frac{1}{2}\left({}^{B_k}V_r - {}^{B_k}V\right)^T M_{B_k}\left({}^{B_k}V_r - {}^{B_k}V\right) \\ & + \frac{1}{2}\sum_{\gamma=1}^{13}\left(\theta_{B_k\gamma} - \hat{\theta}_{B_k\gamma}\right)^2 / \rho_{B_k\gamma} \end{aligned} \tag{29}$$

$$\begin{aligned} \nu_{T_k} = & \frac{1}{2}\left({}^{T_k}V_r - {}^{T_k}V\right)^T M_{T_k}\left({}^{T_k}V_r - {}^{T_k}V\right) \\ & + \frac{1}{2}\sum_{\gamma=1}^{13}\left(\theta_{T_k\gamma} - \hat{\theta}_{T_k\gamma}\right)^2 / \rho_{T_k\gamma} \end{aligned} \tag{30}$$

$$\nu_k = \frac{1}{2} I_{mk}\left(\dot{q}_{kr} - \dot{q}_k\right)^2 + \frac{1}{2}\sum_{\gamma=1}^{4}\left(\theta_{k\gamma} - \hat{\theta}_{k\gamma}\right)^2 / \rho_{k\gamma}. \tag{31}$$

The mathematical operations in [7], [8] lead to

$$\begin{aligned} \dot{\nu}_{mk} = & \ \dot{\nu}_{B_k} + \dot{\nu}_{T_k} + \dot{\nu}_k \\ \leq & \ -\left({}^{B_k}V_r - {}^{B_k}V\right)^T K_{B_k}\left({}^{B_k}V_r - {}^{B_k}V\right) \\ & -\left({}^{T_k}V_r - {}^{T_k}V\right)^T K_{T_k}\left({}^{T_k}V_r - {}^{T_k}V\right) \\ & -k_k(\dot{q}_{kr} - \dot{q}_k)^2 + p_{B_k} - p_{T_k} \end{aligned} \tag{32}$$

where the last two terms represent the *virtual power flows* (in Definition 1) at the two interfaces of the kth module.

When the n modules are connected, it gives

$$p_{T_k} = p_{B_{(k+1)}}, \ k \in \{1, n-1\} \tag{33}$$

which implies

$$\sum_{k=1}^{n}\left(p_{B_k} - p_{T_k}\right) = 0 \tag{34}$$

for given $p_{B_1} = 0$ and $p_{T_n} = 0$.

Thus, the non-negative function for the entire robot is chosen as

$$\nu = \sum_{k=1}^{n} \nu_{mk}. \tag{35}$$

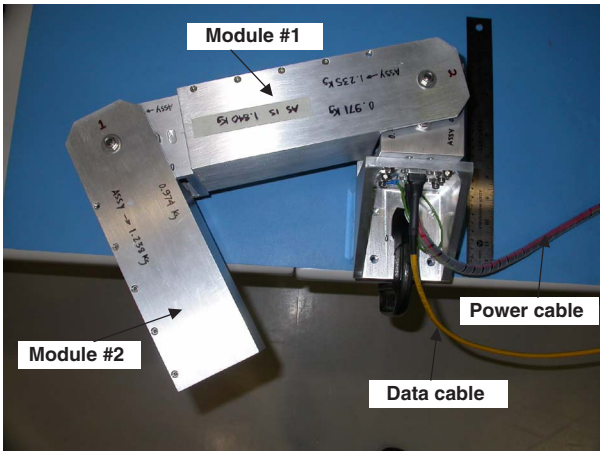

Fig. 5. A two-module assembly.

It follows from (32) and (34) that

$$\begin{aligned}\dot{\nu} \leq & -\sum_{k=1}^{n}\left({}^{B_k}V_r - {}^{B_k}V\right)^T K_{B_k}\left({}^{B_k}V_r - {}^{B_k}V\right) \\ & -\sum_{k=1}^{n}\left({}^{T_k}V_r - {}^{T_k}V\right)^T K_{T_k}\left({}^{T_k}V_r - {}^{T_k}V\right) \\ & -\sum_{k=1}^{n} k_k(\dot{q}_{kr} - \dot{q}_k)^2 \end{aligned} \quad (36)$$

which yields (24) and (25) from (6). The asymptotic stability (26) and (27) follows immediately from bounded reference signals, bounded control τ_k, and bounded joint acceleration $\ddot{q}_k$, $k \in \{1, n\}$, in view of [9].

V. SYSTEM AND TESTING

Two identical modules, as illustrated in Figs. 2 and 3, have been fabricated at the Canadian Space Agency.

Each module weighs about 2.6 Kg. The micro-computer board weighs about 20 grams and the power amplifier weighs about 100 grams. A brushless motor housed together with a harmonic drive (1:100 gear ratio) and an encoder (2000 pulses/rev) is able to produce 11 Nm peak joint torque with an equivalent torque/current constant of 2.6 Nm/A. FPGA-based logic chips are used as embedded computing system. The databus has a speed of 150 Mbps and could be extended to 400 Mbps if needed.

Fig. 5 shows a testbed where the two modules are connected. No control cabinet is presented. The electrical connections are simply two cables (data and power), regardless of the number of modules to be used.

The system has been tested against timing, EMI (electromagnetic interference), and thermal issues. When the two modules are programmed to emulate a robot with 12 virtual modules, the communication time can be well limited to 400 micro-seconds, which allows users to use a sampling frequency of 1000 Hz for communication and control. Outside temperature when working is well within a comfortable zone and no EMI issue has been observed. The system design has been proven to be successful.

VI. CONCLUSIONS

Aimed at enhancing mobility, reformability, flexibility, and extendibility to applications of robot manipulators in unstructured environment without compromising control performance, a novel communication and control scheme has been proposed for modular robotics. Application of the *virtual decomposition control*, supported by a high-speed databus communication system, allows the dynamics issue to be completely handled within individual local modules. This feature permits the host computer to only handle the kinematics related computations. Since all local dynamics based control algorithms are programmed into the embedded micro-computers of the modules, the users could be free from concerns on dynamics issues, while achieving the dynamics-based control performance. The stability of the entire robot is mathematically guaranteed. Two modules have been built. Preliminary tests have demonstrated that the overall design at the system level is very successful. This result enables the VDC based control algorithms to be implemented.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Guangjun Liu at Ryerson University for providing valuable advice on issues at the system level. The authors are also grateful to Joshua Lamorie and Francesco Ricci from Xiphos Technologies Inc. and David Jameux from European Space Agency for providing a high-speed SpaceWire-based communication system with industrial standard.

REFERENCES

[1] T. Fukuda and Y. Kawauchi, "Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator," *Proc. of 1990 IEEE Int. Conf. Robotics and Automation*, pp. 662-667, 1990.

[2] C.J.J. Paredis and P.K. Khosla, "Kinematic design of serial link manipulators from task specifications," *Int. J. Robotics Research*, vol. 12, no. 3, pp. 274-287, 1993.

[3] I.M. Chen and J.W. Burdick, "Determining task optimal modular robot assembly configurations," *Proc. of 1995 IEEE Int. Conf. Robotics and Automation*, pp. 132-137, 1995.

[4] W.K. Chung, J. Han, Y. Youm, S.H. Kim, "Task based design of modular robot manipulator using efficient genetic algorithm," *Proc. of 1997 IEEE Int. Conf. Robotics and Automation*, pp. 507-512, Albuquerque, NM, 1997.

[5] M. Yim, D.G. Duff, and K.D. Roufas, "Polybot: a modular reconfigurable robot," *Proc. of 2000 IEEE Int. Conf. Robotics and Automation*, pp. 514-520, San Francisco, CA, 2000.

[6] S. Lohmeier, T. Buschmann, H. Ulbrich, F. Pfeiffer, "Modular joint design for performance enhanced humanoid robot LOLA", *Proc. of 2006 IEEE Int. Conf. Robotics and Automation*, pp. 88-93, Orlando, FL, 2006.

[7] W.-H. Zhu, Y.-G. Xi, Z.-J. Zhang, Z. Bien, and J. De Schutter, "Virtual decomposition based control for generalized high dimensional robotic systems with complicated structure," *IEEE Trans. Robotics and Automation*, vol. 13, no. 3, pp. 411-436, 1997.

[8] W.-H. Zhu and J. De Schutter, "Adaptive control of mixed rigid/flexible joint robot manipulators based on virtual decomposition," *IEEE Trans. Robotics and Automation*, vol. 15, no. 2, pp. 310-317, 1999.

[9] G. Tao, "A simple alternative to the Barbălat lemma," *IEEE Trans. Automatic Control*, vol. 42, no. 5, p. 698, 1997.