Modular Robot Manipulators with Precision Control

Wen-Hong Zhu*, Tom Lamarche*, Erick Dupuis*, Eric Martin*

*Space Exploration, Canadian Space Agency, Canada
e-mail: Wen-Hong.Zhu@asc-csa.gc.ca

Abstract

Modular robot manipulators have many advantages over conventional integrated robot manipulators, including flexibility, re-configurability, versatility, and low-cost with massive production. However, the lack of control precision has been a long standing drawback in MRM applications until now. In this paper, a VDC-based control and communication system using embedded FPGA (field programmable gate array) logic devices is outlined. This solution is able to push the control precision to a new level without needing joint torque measurements. A test case on a three-module robot using harmonic drives demonstrated that the ratio of the maximum position tracking error to the maximum velocity reached 0.00012 (s).

1 Introduction

Modular robot manipulators (MRMs) have many advantages over conventional integrated robot manipulators. These advantages include flexibility, reconfigurability, versatility, and low-cost with massive production, making MRMs particularly useful for applications in unstructured environments such as space where human access is severely limited.

Modular robotics has been extensively studied in the past. An early type MRM was reported in [1]. Fukuda *et. al.* introduced the concept of cellular robotics [2]. Paredis and Khosla developed a reconfigurable modular manipulator system (RMMS) with modular links and joints [3]. Using an assembly incidence matrix (AIM) representation, Chen and Burdick developed a (kinematic) configuration optimization procedure [4]. A genetic algorithm leading to task-based design was proposed by Chung *et. al.* [5]. Yim *et. al.* developed a reconfigurable modular robot named PolyBot targeting space applications [6]. Hirzinger's group at DLR developed a light weight robot (LWR) with mechanical modularity [7]. Shen *et. al.* developed a SuperBot modular robot using an automatic docking mechanism [8]. Recently, a MRM using joint torque sensors was developed in [9, 10]. Li *et. al.* proposed a decentralized robust control resulting in a 50% performance improvement over PID control [11].

However, the advantages of MRMs come with a price. The decentralized mechanical, electrical, and computing structures prevent many centralized control approaches from being directly applied [9, 12]. As a result, only very limited communications among modules are permitted such that either very simple local controllers (such as a PID controller) are used, resulting in severely limited control performance and precision, or joint torque sensing approaches are used [9, 10].

The joint torque sensing approach basically converts the computational aspect of robot link dynamics to a measurement mean. Besides introducing additional flexibility to the joints, which might be undesirable for control bandwidth, the joint torque measurements need to be directly used to compensate for the robot dynamics to achieve asymptotic trajectory tracking control. While being tested very successfully for a robot in free motion [10], the direct use of joint torque measurements for robot dynamics compensation would necessarily create *algebraic loops* (see Section 2.10 in [13]) that make robotic systems sensitive to unmodeled dynamics and disturbances and might even result in instability when the gain of an *algebraic loop* reaches *one*.

The *virtual decomposition control* (VDC) provides an effective solution to this long-standing problem without needing joint torque sensing [13]-[18]. The VDC approach uses subsystem (such as links and joints of a complex robot) dynamics to conduct control design, while rigorously guaranteeing the stability and convergence of the entire robot manipulator. The amount of information exchange among modules is very limited. The central concept of VDC is the definition of the *virtual stability*. The stability of the entire robot manipulator is mathematically equivalent to the *virtual stability* of every module. This fact allows us to convert a large problem to a few easy-to-solve simple problems with mathematical certainty [13].

In this paper, a three-module MRM is presented, see Fig. 1, using VDC [18] and embedded FPGA (field programmable gate array) logic devices [19, 20]. A

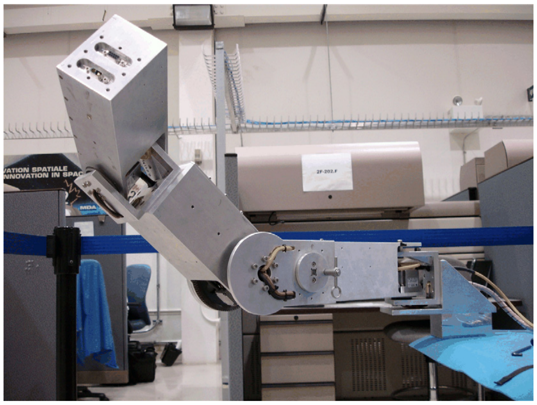

Figure 1. A three-module assembly.

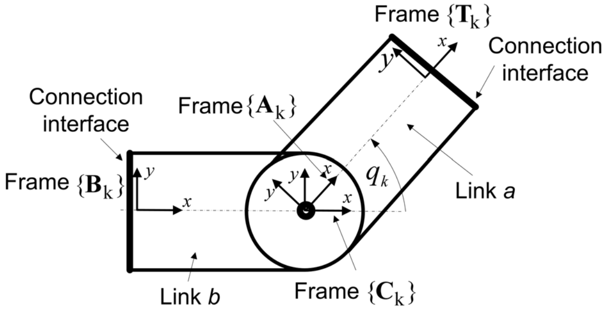

Figure 2. Coordinate frames for the kth module.

hierarchical master-slave structure is used to compute VDC, supported by a high-speed data bus which is a modified version of SpaceWire (IEEE 1355) by Xiphos Technologies Inc. The FPGA logic computation allows ultra-fast state update rates up to the clock rate (50+ MHz). These fast update rates make the control performance of FPGA logic control systems approach what is possible for their analog counterparts, while keeping all advantageous features of digital systems.

This paper is organized as follows: The advantages of using this type of MRMs in space applications are discussed in Section 2. A three-module MRS with gravity counterbalance capability is formally presented in Section 3. Control and communication are outlined in Section 4. Finally, an experimental result is presented in Section 6 together with a performance comparison to other approaches, followed by conclusion.

2 Advantages of Using MRMs

Compared to state-of-the-art space robot manipulators, the advantages of using MRMs with VDC and FPGA logic devices can be summarized as follows:

i) Embedded control electronics such as FPGA logic devices allows the use of ultra-high sampling rate (in a range of Megahertz) to properly handle uncertain dynamics at joints, leading to effective friction compensation necessarily needed for precision control [20].

ii) Dynamics of each module can be handled by local embedded controller, allowing central computer to focus on kinematics computation only.

iii) MRMs have serial communication among modules, eliminating the use of excessive amount of cabling often seen in most integrated robot manipulators.

iv) The integration of electronics into mechanical assembly makes the MRMs particularly suitable for mobile applications. Without requiring additional space for its electronics and control cabinet, a MRM can be conveniently mounted on any mobile platform as long as the power and data connectors are provided.

v) MRMs allow on-site change of robot structure and even the change of motion degrees of freedom by adding/removing modules to/from the robot.

vi) It makes easy repair possible, since a defective module can be easily replaced.

vii) Mass production of modules will eventually reduce the cost of using MRMs. This is an effective answer to the tight budget situation the entire space community is facing.

3 A Three-Module MRM

The three-module MRM as illustrated in Fig. 1 uses two types of modules, namely type-I and type-II, respectively. The differences of a type-II module from a type-I module are the use of a counter-balance spring for compensating the gravity [21] and the use of a bigger motor [18]. The first and the third modules are two type-I modules, and the second module is a type-II module. Each module consists of two links connected by one joint, see Fig. 2. A pair of male/female connection interfaces is located at both ends providing power and data connections.

Four coordinate frames are attached to each module. For the kth module, $k \in \{1, 3\}$, the four frames are shown in Fig. 2. Frames $\{\mathbf{A}_k\}$ and $\{\mathbf{C}_k\}$ share the same origin and are attached to Link a and Link *b*, respectively. The $\mathbf{z}$ axes of the two frames coincide with the joint axis. Frames $\{\mathbf{T}_k\}$ and $\{\mathbf{B}_k\}$ are attached to the two connection interfaces of the *k*th module, respectively, with their $\mathbf{x}$ axes aligning with the two link axes.

The mechanical parameters of the two types of modules are listed in Table 1.

4 Control and Communication

The VDC approach [13, 18] is an effective approach to precision control of MRMs. This approach allows embedded computations for module dynamics, permitting the master computer for kinematics-only computation with guaranteed stability [17]. The entire control computation uses a model-based *virtual system* with numerically available *required* variables - the variables that are required to have the (position and/or force) control objectives accomplished.

The technical merit of the VDC approach is its way of handling dynamic interactions among the modules. These dynamic interactions are entirely represented by the *virtual power flows* (VPFs) that are defined as the inner products of the velocity errors and the force errors between the virtual and physical systems. The use of VPFs leads to the definition of the *virtual stability* in [13]. The *virtual stability* of every module (combined with its embedded control) is mathematically equivalent to the stability of the entire MRM.

The VDC approach for precision control of MRMs can be summarized into six steps:

1) Assign appropriate frames to all the modules.

2) Make velocity mappings from the independent physical and required velocity coordinates to all physical and required velocities at the body frames.

3) Compute the required net force/moment vectors for all links.

4) Propagate all required force/moment vectors from the robot tip to the base.

5) Compose joint control torques for all module joints.

6) Conduct stability analysis.

The detailed VDC algorithm for this type of MRMs has been thoroughly presented in [18].

The VDC-based communication system is illustrated in Fig. 3, which is using a modified version of the SpaceWire bus clocked at 150 Mbps. The detailed implementation was described in [18, 22]. The master node is a Xilinx Virtex-II-Pro FPGA configured PowerPC 405 clocked at 300 MHz. Each of the three slave nodes uses a Xilinx Virtex-II-1000 FPGA logic device clocked at 50 MHz. The master node (programmed in C) controls the timing for data communication with a sampling period of 1 millisecond, corresponding to a 1000 Hz sampling rate.

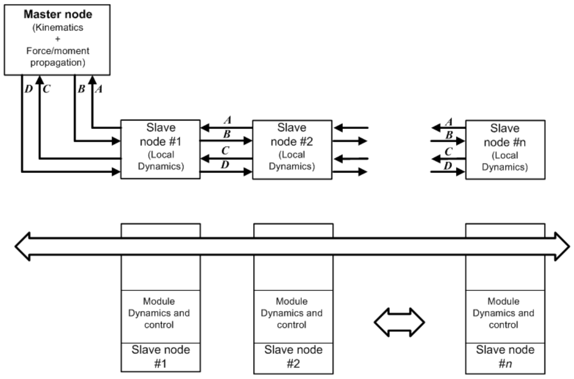

Figure 3. Communication scheme and data flow.

As shown in Fig. 3, the communication protocol comprises two cycles with four actions (namely *A*, *B*, *C*, and *D*) within each sampling period. The first communication cycle constitutes *Communications A* and *B*, and the second communication cycle constitutes *Communications C* and *D*. To meet the 1 ms sampling period, the entire communication process is synchronized at *Communication D*. In each communication action, a data package of 40 bytes is transferred between the master node and each slave node.

Xilinx graphical programming tool *System Generator* is used to perform FPGA logic control programming for each slave node. This tool needs to be embedded on MATLAB Simulink and has limited elementary block selection. At present stage, no division block is available.

The sequential process for the control and communication is as follows:

Step 1) *Communications A*: the master node extracts the joint position and velocity of each slave node.

Step 2) The master node computes the required joint velocities and the actual and required linear/angular velocity vectors in frame $\{\mathbf{C}_k\}$ for $k = 1, 2, 3$ from the robot base toward the robot tip.

Step 3) *Communications B*: the master node sends the required joint velocity and the actual and

Table 1. Module mechanical parameters

Type-I module			Type-II module		
Weight	2.6 (kg)		Weight	4.9 (kg)	
Actuator	FHA-11C-100		Actuator	FHA-14C-100	
Peak torque	11 (Nm)		Peak torque	28 (Nm)	
Continuous torque	4.2 (Nm)		Continuous torque	6.8 (Nm)	
Encoder pulse	2000		Encoder pulse	2000	
Gear ratio	100		Gear ratio	100	
Spring torque	-		Spring torque	<11 (Nm)	
Link parameters*	Link a	Link b	Link parameters*	Link a	Link b
m (kg)	1.8	0.5	m (kg)	3.0	1.64
mr_x (kgm)	0.135	-0.0175	mr_x (kgm)	0.108	-0.13
mr_y (kgm)	0	0	mr_y (kgm)	0	0
mr_z (kgm)	-0.0216	-7.5×10^{-3}	mr_z (kgm)	-0.012	0
mr_x^2 (kgm^2)	0.01	4.5×10^{-4}	mr_x^2 (kgm^2)	3.9×10^{-3}	0.01
mr_y^2 (kgm^2)	0	0	mr_y^2 (kgm^2)	0	0
mr_z^2 (kgm^2)	2.59×10^{-4}	1.13×10^{-4}	mr_z^2 (kgm^2)	4.8×10^{-5}	0
$mr_xr_y - I_{xy}$ (kgm^2)	0	0	$mr_xr_y - I_{xy}$ (kgm^2)	0	0
$mr_xr_z - I_{xz}$ (kgm^2)	-2×10^{-3}	2.63×10^{-4}	$mr_xr_z - I_{xz}$ (kgm^2)	-4.32×10^{-4}	0
$mr_yr_z - I_{yz}$ (kgm^2)	0	0	$mr_yr_z - I_{yz}$ (kgm^2)	0	0
I_{xx} (kgm^2)	2×10^{-3}	8×10^{-4}	I_{xx} (kgm^2)	4×10^{-3}	3.6×10^{-3}
I_{yy} (kgm^2)	8.5×10^{-3}	1×10^{-3}	I_{yy} (kgm^2)	0.014	6×10^{-3}
I_{zz} (kgm^2)	6.5×10^{-3}	5×10^{-4}	I_{zz} (kgm^2)	0.01	3×10^{-3}

* The 13 parameters of each link are defined in Appendix A of [13]. In principle, only 10 of the 13 parameters are independent. However, they are all needed for VDC computation. The parameters of Link a are referenced to frame $\{\mathbf{A}_k\}$ and the parameters of Link b are referenced to frame $\{\mathbf{C}_k\}$ for the kth module.

required linear/angular velocity vectors in frame $\{\mathbf{C}_k\}$ to each slave node, respectively.

Step 4) Each slave node constructs the actual and required linear/angular velocity vectors in frame $\{\mathbf{A}_k\}$ and, then, computes the required net force/moment vectors of the two links. This is a decentralized computational process.

Step 5) *Communications C*: the master node extracts the required force/moment vectors of the two links from each slave node.

Step 6) The master node computes the required force/moment vector in frame $\{\mathbf{A}_k\}$ for $k = 3, 2, 1$ from the robot tip toward the robot base.

Step 7) *Communications D*: the master node sends the projection of the required force/moment vector onto the joint to each slave node.

Step 8) Each slave node computes the joint control torque using both the joint dynamics and the projection of the required force/moment vector onto its joint (the link dynamics received from the master node). Again, this is a decentralized computational process.

The stability of the control has been rigorously proven in [18], as well as in Chapter 12 of [13].

5 Experiments

A series of experiments have been conducted on this system. An experimental result is presented below.

5.1 Experimental Result

The test trajectories for the three joints are designed as

$$q_{1d}(t) = q_d(t) \quad (1)$$

$$q_{2d}(t) = 2q_d(t) \quad (2)$$

$$q_{3d}(t) = 3q_d(t) \quad (3)$$

with

$$\begin{aligned} q_d(t) &= 6(p_f - p_0)\,(t/t_f)^5 - 15(p_f - p_0)\,(t/t_f)^4 \\ &\quad +10(p_f - p_0)\,(t/t_f)^3 + p_0 \end{aligned} \quad (4)$$

where $p_0 = 0$ rad, $p_f = 0.1571$ rad, and $t_f = 1$ s denote the initial position, terminal position (in which 0.1571 rad corresponds to 20,000 encoder counts),

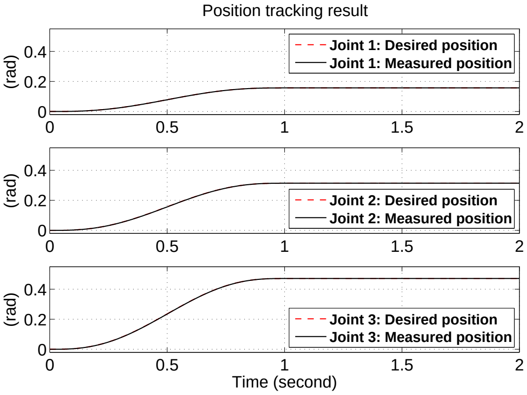

Figure 4. Position tracking result of the 3-module robot.

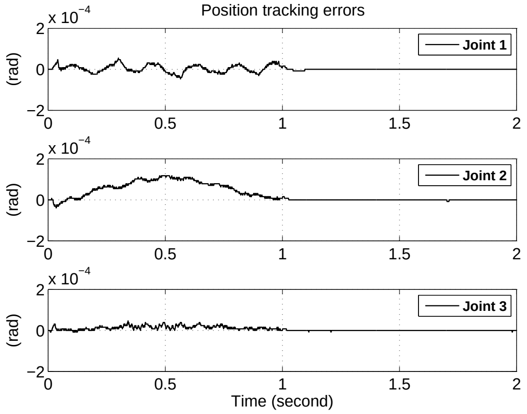

Figure 5. Position tracking errors.

and the trajectory duration, respectively. This gives $\max(\dot{q}_d) = (15/8)(p_f - p_0)/t_f = 0.2946$ rad/s and $\max(\ddot{q}_d) = (45/8)(p_f - p_0)/t_f^2 = 0.8837$ rad/s^2.

The position tracking result and the tracking errors of the three joints are shown in Figs. 4 and 5, respectively.

5.2 Performance Comparison

A performance indicator is given below

$$\rho = \min_{\forall \iota(p(t)) \in \mathbb{W}} \frac{\max \|\mathbf{e}\|}{\max \|\mathbf{v}\|} \tag{5}$$

where $\iota(p(t))$ represents a trajectory either in joint space or in Cartesian space with $p(t) \in [0, 1]$ being a trajectory planning factor, $\mathbb{W}$ represents the overall trajectory space, and $\mathbf{e}$ and $\mathbf{v}$ represent the position tracking error and velocity vectors in SI units, along the trajectory ι.

It is the authors' opinion that index ρ defined by (5) characterizes the tracking control performance of a given controlled robot. The observation behind this performance indicator is that a larger velocity is often associated with a larger acceleration within a given operational space, resulting in a larger position tracking error subject to given dynamic uncertainty. Thus, the minimum ratio of the maximum position error to the maximum velocity reflects the performance of a controller in a comparative manner.

Table 2 lists the ratios of the maximum position tracking error to the maximum velocity for a few modular robot manipulators using harmonic drives. The first four rows correspond to four modular/integrated robots in existing publications. The last three rows correspond to the experimental results of the studied MRM. These three rows represent PID control with a 1000 Hz sampling rate (where control gains are reduced to maintain the stability), PID control with embedded FPGA, and VDC with embedded FPGA, respectively.

In Table 2, the advantage of using VDC can be clearly seen by comparing row 6 with row 7, and the advantage of using embedded FPGA implementation can be clearly seen by comparing row 5 with row 6 and row 4 with row 7.

6 Conclusions

A modular robot manipulator capable of performing precision control without needing joint torque sensors has been presented in this paper. Using VDC supported by a high-speed databus (a modified version of SpaceWire), the control system allows each slave computer to fully handle its local module dynamics and the master computer to handle the kinematics with guaranteed stability. The use of FPGA logic devices as embedded slave controllers permits ultra-fast state update, leading to effective joint friction compensation necessarily needed for precision control. The precision control performance has been verified through experiments. The ratio of the maximum position tracking error to the maximum velocity reached 0.00012 (s).

References

[1] D. Schmitz, P. Khosla, and T. Kanade, "The CMU reconfigurable modular manipulator system," Institute for Software Research, Carnegie Mellon University, 1988.

[2] 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.

Table 2. Performance comparison

	Systems	Maximum velocity	Maximum error	Ratio of error to velocity
Modular/integrated robots in previous publications	MRM in [5]	0.33 rad/s	0.006 rad	0.01818 s
	MRR in [11]	80.78 deg/s	1.35 deg	0.01671 s
	MRR in [10]	47 deg/s	0.15 deg	0.00319 s
	VDC with 1000 Hz [23]	0.314 rad/s	0.0005 rad	0.00159 s
Modular robot developed in this paper	PID with 1000 Hz	0.5226 m/s	2.3 mm	0.00440 s
	PID with embedded FPGA	0.5226 m/s	0.26 mm	0.00050 s
	VDC with embedded FPGA	0.5226 m/s	0.0618 mm	**0.00012 s**

[3] 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.

[4] 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.

[5] W. K. Chung, J. Han, Y. Youm, and 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.

[6] 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.

[7] G. Hirzinger, A. Albu-Schaffer, M. Hahnle, I. Schaefer, and N. Sporer, “On a new generation of torque controlled light-weight robots,” *Proc. of 2001 IEEE Int. Conf. Robotics and Automation*, pp. 3356-3363, Seoul, Korea, 2001.

[8] W.-M. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein, and J. Venkatesh, “Multimode locomotion via SuperBot reconfigurable robots.” *Autonomous Robots*, vol. 20, no. 2, pp. 165-177, 2006.

[9] G. Liu, S. Abdul, and A. A. Goldenberg, “Distributed control of modular and reconfigurable robot with torque sensing,” *Robotica*, vol. 26, no. 1, pp. 75-84, January 2008.

[10] G. Liu, Y. Liu, and A. A. Goldenberg, “Design, analysis, and control of a spring-assisted modular and reconfigurable robot,” *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 4, pp. 695-706, 2011.

[11] Z. Li, W. W. Melek, and C. Clark, “Decentralized robust control of robot manipulators with harmonic drive transmission and application to modular and reconfigurable serial arms,” *Robotics*, vol. 27, pp. 291-302, 2009.

[12] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. Chirikjian, “Modular self-reconfigurable robot systems,” *IEEE Robotics and Automation Magazine*, pp. 43-52, March 2007.

[13] W.-H. Zhu, *Virtual Decomposition Control-Toward Hyper Degrees of Freedom Robots*, Berlin Heidelberg, Germany: Springer-Verlag, 2010.

[14] 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.

[15] W.-H. Zhu and J. De Schutter, “Control of two industrial manipulators rigidly holding an egg,” *IEEE Control Systems Magazine*, vol. 19, no. 2, pp. 24-30, 1999.

[16] W.-H. Zhu and J. De Schutter, “Experimental verifications of virtual decomposition based motion/force control,” *IEEE Trans. Robotics and Automation*, vol. 18, no. 3, pp. 379-386, 2002.

[17] W.-H. Zhu and T. Lamarche, “Modular robot manipulators based on virtual decomposition control,” *Proc. of 2007 IEEE Int. Conf. Robotics and Automation*, pp. 2235-2240, Rome, Italy, April 2007.

[18] W.-H. Zhu, T. Lamarche, E. Dupuis, D. Jameux, P. Barnard, and G. Liu, “Precision control of modular robot manipulators: The VDC approach with embedded FPGA”, *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1162-1179, Oct. 2013.

[19] E. Monmasson and M. N. Cirstea, “FPGA design methodology for industrial control systems - a review,” *IEEE Trans. Industrial Electronics*, vol. 54, no. 4, pp. 1824-1842, 2007.

[20] W.-H. Zhu, "FPGA logic devices for precision control: An application to large friction actuators with payloads," *IEEE Control Systems Magazine*, vol. 34, no. 3, June 2014.

[21] W.-H. Zhu, T. Lamarche, and P. Barnard, "Modular robot manipulators with preloadable modules," in Invited Session on Modular Robotics with *2007 IEEE Int. Conf. Mechatronics and Automation*, pp. 7-12, Harbin, China, August 2007.

[22] T. Lamarche and W.-H. Zhu, "A virtual decomposition control based communication network for modular robots applications," *First International Workshop on Networking Technology for Robotics and Applications (NeTRA 2007) in conjunction with ICCCN 2007*, pp. 1321-1326, Honolulu, Hawaii, USA, August 2007.

[23] W.-H. Zhu, E. Dupuis, and M. Doyon, "Adaptive control of harmonic drives," *ASME J. Dynamic Systems, Measurement, and Control*, vol. 129, pp. 182-193, March 2007.