High-speed Communication Network for Controls with the Application on the Exoskeleton

Sunghoon Kim*, George Anwar** and H.Kazerooni***

Abstract— **Our lower extremity exoskeleton is a wearable robotic device that enables a human to walk with a load for a prolonged period of time without reducing the human's agility. The exoskeleton comprised of two anthropomorphic legs and a spine. The device is designed and controlled autonomously via a spine mounted internal combustion engine and PC104 compliant computer. Custom hardware was designed to accommodate the high-speed network required to link the distributed sensors and actuators. This paper presents the implementation of the high-speed hard real-time network designed to maintain stable control of the exoskeleton during stance and walk. The high-speed ring protocol necessary to maintain strict hard real-time synchronization between the distributed sensors and actuator of the exoskeleton is presented. Communication latency was considered in the design with respect to its impact on performance, and stability. Error detection and recovery was crucial for operation with the exoskeleton. A *Cyclic Redundancy Check* (CRC) algorithm was incorporated into the protocol to achieve this error detection. The use of this high-speed serial based network greatly minimized the number of cables required over traditional parallel-based systems. The achieved update time of up to 10 kHz for a six-actuator system enables the architecture to be viable system for most industrial controls.**

I. INTRODUCTION

The function of the lower extremity exoskeleton is to off load the weight of the payload from its occupant while walking. One method to achieve this functionality is to minimize the force between the human and the machine. The exoskeleton consists of sixteen (16) accelerometers, ten (10) encoders, eight (8) foot sensors, six (6) force sensors, one (1) load cell, one (1) inclinometer, and six (6) hydraulic servo valves. The required control system for the exoskeleton consists of a highly coupled multi-input multi-output dynamics, which precludes the effective use of a decentralized control system. The over 200 wires required to carried the various sensors and actuation signals forced us to design a networked architecture. In turn, the communication system of this architecture must be able to transfer a large number of data instantly to reduce data latency. As a result a general purpose high-speed serial communication network using a ring topology (ExoRing) and an optimized ring protocol (ExoRing Protocol) to transfer desired data effectively was developed. This enabled the exoskeleton control system to deliver all of the necessary data to and from the central control unit with only 24 wires.

There has been a great deal of interest in ring networks and they are used in many applications [1],[2],[3]. Stations in ring networks are connected by serial point-to-point links in a circular fashion. The developed ExoRing protocol is different from the commonly used token ring protocol since there is no token circulating around the ring, instead, an actuation data packet is recognized as a token that grants the right of access to the network and all actuation data assigned for the stations on the ring are transferred all together. That is, the conventional ring protocol only grants one token to a certain station with high priority so after it receives the token, it releases its packets onto the ring [1], but in the ExoRing protocol more than two stations could be allowed to transmit packets when they receive their actuation data and there is no congestion or collision on the ring. The protocol is designed to transfer predefined data for each station and optimized for the deterministic hard real-time transmission. Ring topologies for real-time control have been investigated and used in industry [4], [5]. The whole communication protocol in this distributed communication system will also be discussed and how the sampling period and the control feedback loop frequency have effects on the performance in sense of phase margin loss caused by the serial communication

The paper is structured as follows. Section 2 presents our own unique Exoskeleton Communication Architecture that realizes the serial ring topology and is specially designed for the Exoskeleton. Section 3 will discuss about the communication considerations including the communication goal. Section 4 will introduce the communication elements defined in this paper, our designed protocol and the basic communication algorithm using the communication elements to fulfill the requirement, and the estimated communication time is calculated at the end. In section 5, there will be an analysis on the communication, and the estimated and measured communication time and phase latency are also present. Finally, section 6 and 7 will give future works and a short conclusion about the paper.

II. EXOSKELETON COMMUNICATION ARCHITECTURE

A. Overall Communication Architecture

The Exoskeleton communication architecture consists of the ExoBrain that manages the whole communication net-

*Sunghoon Kim is a Graduate Student of Mechanical Engineering, University of California, Berkeley. shkim@me.berkeley.edu

**George Anwar is President of Integrated Motions, Inc. ganwar@integratedmotions.com

***H.Kazerooni is a Professor of Mechanical Engineering, University of California, Berkeley. kazerooni@me.berkeley.edu

0-7803-8335-4/04/$17.00 ©2004 AACC

work, Remote I/O modules (RIOMs), and the GUI interface module. The ExoBrain as a central computing unit for the control and a central supervising unit for the network of the Exoskeleton is made up of three PC/104-Plus compliant stackable boards: ExoCPU, ExoPCI and Supervisory I/O modules (SIOMs) board. All computations related to the control algorithm requiring complex equations are performed in the ExoCPU. The ExoPCI is an interface unit between the ExoCPU and SIOMs to make use of the PC/104-plus interface with the ExoCPU and parallel bus interface with the SIOMs. There exist three (SIOMs) on the SIOM board. Each SIOM can be linked to a maximum of seven RIOMs. The high-speed serial ExoRing is established via each channel of a SIOM, for example, the data flows from SIOM0 to RIOM0 to RIOM1 to RIOM2 back to SIOM0. A loop-back terminator must be plugged into the transmit port of the last RIOM on each ExoRing to complete the ring. The overall communication architecture is illustrated in Fig. 1. Each SIOM can have the connection to their RIOMs

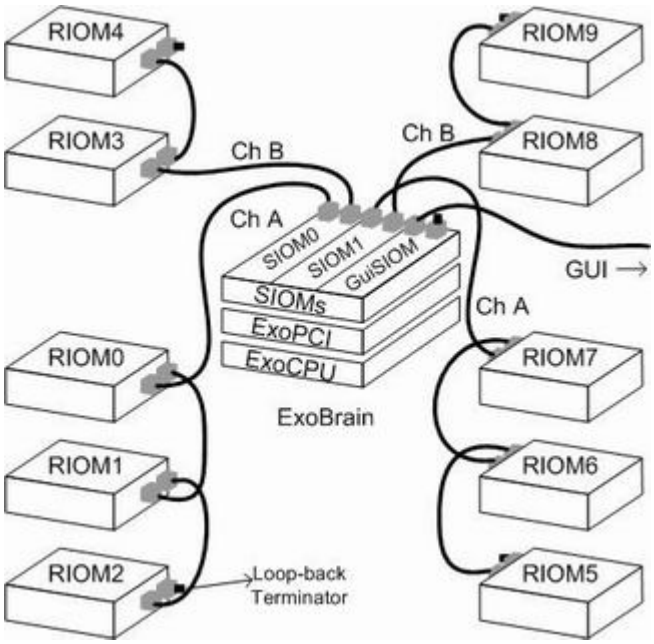

Fig. 1. Overall Exoskeleton communication architecture

either through one channel (One-channel communication) or through two channels (Two-channel communication). The Exoskeleton uses Two-channel communication. The choice of Two-channel communication was to minimize the length of the communication cable. Cable routing was also simplified. Fig. 1 illustrates 4 ExoRings: one for 3 RIOMs (RIOM0, RIOM1 and RIOM2) located on the right leg of the Exoskeleton, 3 RIOMs (RIOM5, RIOM6 and RIOM7) on the left leg, 2 RIOMs (RIOM3 and RIOM4) on the right side of the back, and 2 RIOMs (RIOM8 and RIOM9) on the left side of the back. 2 SIOMs configured for Two-channel communication are used to establish the 4 ExoRings, 1 SIOM for the left and 1 SIOM for the right. A third SIOM (GuiSIOM) is used to communicate with a graphical user interface (GUI). The GUI system is a desktop PC hosting a LabView based GUI program for the sole purpose of monitoring system parameters and interactive tuning of the exoskeleton.

B. HotLink Transceiver

A 200Mbaud Cypress CY7C924ADX Hotlink[6] transceiver was selected for the communication building block. This transceiver is the point-to-point communication block allowing data transfer over high-speed serial links (optical fiber, balanced, and unbalanced copper transmission lines - Firewire in the Exoskeleton). The Hotlink devices are designed for a variety of applications where parallel interfaces can be replaced with high-speed, point-to-point serial links. This transceiver also supports daisy-chain topology. Basically, when it transmits an outgoing data, it receives a parallel 8-bit data from its control host and stores it into its Transmit FIFO. The data is encoded into a 10-bit parallel data, serialized as bitstream and released into its down-stream through its transmit port (TX port) at the character rate, 10 times faster than the reference clock that is used for parallel data. On the contrary, when it receives an incoming data through its Receiver port (RX port), the 10-bit bitstream data is deserialized into a 10-bit parallel data, decoded into 8-bit parallel data, and stacked onto a Receive FIFO. The control host in turn can read the data whenever ready and before the Receive FIFO is full.

C. ExoRing topology

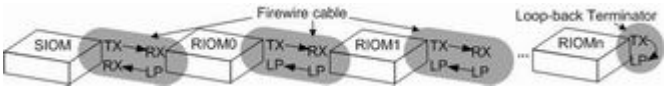

Fig. 2. ExoRing topology

Fig. 2 illustrates how the ExoRing topology is accomplished between the SIOM and the n+1 RIOMs communication. A serial Firewire cable consists of 6 wires: $\pm$ power and 2 $\pm$ differential pair communication wires. These differential pairs are used for the high-speed transmission in both directions : TX and RX. To build the ExoRing topology using the transceiver, the loop (LP) lines are introduced in the RIOM as shown in Fig. 2. The path of the ring topology is hence the TX of the SIOM $\rightarrow$ the RX of the RIOM0, the TX of the RIOM0 $\rightarrow$ the RX of the RIOM1, and so on. This process is repeated until a path reaches the RX of the RIOMn. The path is completed by a loop-back terminator. The loop-back terminator is a connector with both differential lines of the same polarity shorted.

This connector on a RIOM is designed such that the ring topology is complete without the need to completely circulate the cable. There needs to be only one communication cable between any two RIOMs and a loop-back connector on the last RIOM on the ring.

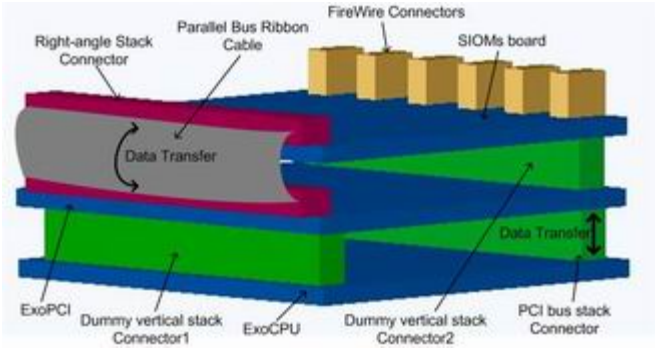

Fig. 3. ExoBrain structure

D. ExoBrain

As shown in Fig. 3, the Exobrain consists of the SIOMs board, the ExoPCI and the ExoCPU. The SIOMs board contains 3 SIOMs, and each SIOM has one transceiver and one FPGA control unit. These control units are connected to the logic unit on the ExoPCI via the 64-bit Parallel bus ribbon cable. All data transfer between the ExoPCI and three SIOMs occur through this cable. The ExoPCI is an PCI interface unit that utilizes the PC/104-plus interface to access the main memory of the ExoCPU. This memory acts as a storage of all data. This PCI interface performs 32-bit data transfer at 33Mhz.

The primary function of each ExoBrain sub-components can be summarized as follows:

- ExoCPU : performs control equations computation and provides the system memory with new actuation data at each control loop iteration.
- ExoPCI : transfers actuation data from system memory to the SIOM0 and SIOM1; transfers sensor data from SIOM0 and SIOM1 to system memory.
- SIOM0 and SIOM1 : transfers actuation data from the ExoPCI to the RIOMs; transfers sensor data from the RIOMs to the ExoPCI.

E. RIOM

The RIOMs are strategically distributed throughout the Exoskeleton. Their task is to solely perform data acquisition from all the sensors and provide drive signal to all the actuators. Each RIOM has three A-to-D converters (ADCs), one D-to-A converter (DAC), one quadrature decoder, and an eight bit digital input port.

The on board ADCs are 16-bits and capable of 85 ksps. These ADCs are used to read the accelerometers, inclinometer, force sensors, and load cell on the Exoskeleton. The digital inputs are used to read 8 foot switches. The quadrature decoder is used to decode each incremental encoder of each joint. In addition, the RIOM performs digital differentiation to computes the velocity from the joint angle data. Each RIOM is capable of converting 6 sensor data and 1 actuator command. There is one transceiver and one FPGA control unit on each RIOM.

III. Communication Consideration

The main purpose of the communication is to achieve fast, accurate, and real-time system data transfer without loss of data and any unnecessary latency. The Exoskeleton control system requires 10 RIOMs which results in a transfer of 70 pieces of data every control cycle. Designing for a maximum control loop frequency of 10 kHz required 0.7 million data transfers every second. To maintain control stability, the data must be transferred instantaneously, or in a very small fraction of the control loop time. The communication time is defined as time between transfer of the actuator commands from system memory to the receipt of all sensor data to system memory. 20% of the control loop time was arbitrarily chosen as a design target for communication. The target control loop time was selected to be 10 kHz. Therefore, 20 μs is set aside for communication while the remaining 80 μs is allocated for control algorithm computation by the ExoCPU. Fig. 4 illustrates the time allocation design goals for the serialized control network.

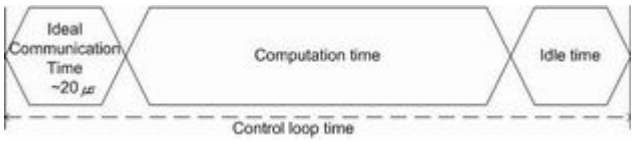

Fig. 4. Control Loop Time Distribution

IV. Communication Protocol

A. Packet Format

A packet is the basic element of a message and contains one valuable information. The information may either be actuator command, sensor data, system command, or error message. The size of actuator or sensor data are 16-bits. The data is augmented by 8-bits to create a 24-bit packet format. The packet format is designed as shown in Fig. 5. The

Fig. 5. Basic Packet Format

augmented 8-bits are divided for 2-bits to determine the type of packet (TYPE), 3-bits for the address of the RIOM on the ring, and a 3-bits CRC algorithm for error detection. The four possible packet types are software command (CMD), actuator command DAC , sensor data (DAT), and error (ERR). The 3-bits address allows for up to eight devices on any single ring (one SIOM and seven RIOMs). The 3-bits CRC is a result of an algorithm on the 21-bits of the packet made up of the type, address and data bits. Each packet has its own CRC. This allows the receiver to detect an error as soon as a corrupted packet arrives. 3-bits CRC are designed to detect all one-bit, most of two-bits and burst error composed of series of 11...11. The best CRC polynomial in 3-bits is either 1011 or 1101 in this case but

both of them miss detection of data contaminated by two-bit errors of multiple of 10000001 such as 0...0100000010...0. That is, this CRC logic can't detect data whose two bits having 6 bits distance between them are flipped. There are 21 cases of error undetection among 210 cases of two-bit errors in 21-bit data. According to the specifications of the Cypress CY7C924ADX Bit Error Rate (BER) is much less than 1x10-12 [7]. Statistically at 10kHz this means that it misses detection of a corrupted packet with approximately 2.1×10^{-21}% possibility, which is 1 missed detection in over 2 billions year of continuous operation if two bit errors are independent.

B. Message Format

Each RIOM assembles 6 data packets and appends them to its actuator command packet into a single message every sample period. Each message is initiated with a Start-Of-Message code (SOM) and terminated with an End-Of-Message code (EOM). The message format is shown in Fig. 6. This message format provides the flexibility to make

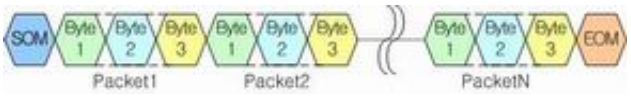

Fig. 6. Message format

up messages with any number of packets. The number of packets within a message is limited by the tolerable communication latency of the control algorithm.

C. Data Transfer

The fundamental data transfer of the entire exoskeleton is illustrated in Fig. 7.

One actuator command data and six sensor data are associated with each RIOM. DAC# is the actuator command packet for RIOM#. DAT#% represents the %+1th sensor data from RIOM#. The data transfer starts with each SIOM's transmission of the all the DAC packets. Each RIOM, in turn appends its corresponding six sensor data packets to the message as it receives its actuator command packet. The augmented message is transmitted to the next RIOM on the ring. This process continues through the last RIOM in the ring which transmit the entire message chain back to the SIOM.

V. Communication Analysis

Communication latency due the transfer of system data across a serially distributed system can have a profound effect on the stability and performance of a control system versus a conventional parallel system. In the following subsections a detail breakdown of the latency of the current ExoRing is discussed.

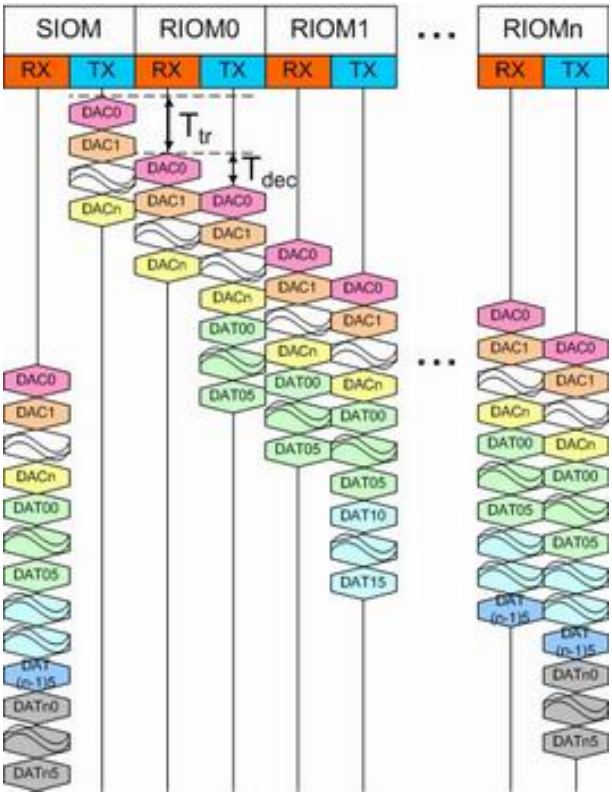

Fig. 7. Data transfer on the ring

A. Estimated Communication Time and experimental results

The communication time is based on a 20 Mhz reference clock and the 33 Mhz PCI clock. Fig. 8 illustrated the timing diagram for a single control loop cycle.

The time parameters shown in Fig. 8 can be categorized as follows:

- T_1 : Time to read DACs from memory via the PCI bus
- T_2 : Time to send DACs from the ExoPCI to the SIOMs
- T_3 : Time to send DACs and receives DATs onto the ExoRing
- T_4 : Time to send the last DAT data from the SIOMs to the ExoPCI
- T_5 : Time to send data to memory via the PCI bus

T_1 results in n+16 clocks at the 33Mhz PCI clock. Where n is the number of 32-bit data sent across the PCI bus. In the case of the Exoskeleton, the total transfer is 8 32-bit words resulting a total T_1 of 24 clocks (0.72μs at 33 Mhz).

T_2 is m+1 clocks, m clocks for m DACs transfer across the parallel data bus and 1 clock for a cycle to terminate the transaction at the reference clock of 20 Mhz. For the current Exoskeleton T_2 is 8 clocks (0.40μs).

T_3 can be calculated for the one-channel or the two-channel communication case according to the following

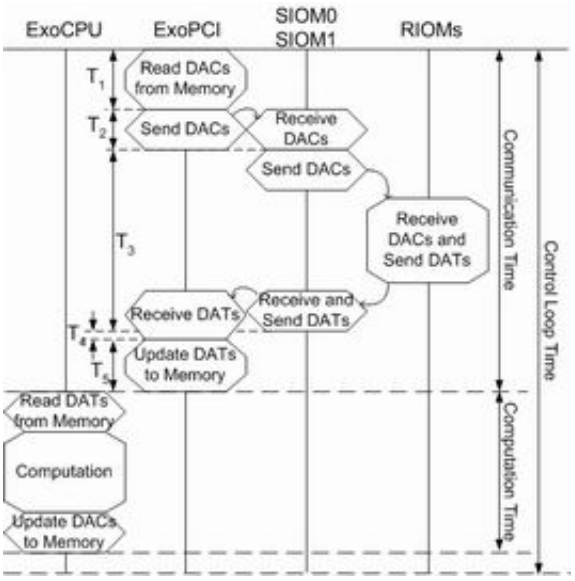

Fig. 8. Overall Communication Timing Diagram

equations:

$$T_3 = (N_{RIOM}+1)\times(T_{tr}+T_{dec}+T_{under})+ N_{RIOM}\times(N_{DAC}+N_{DAT})\times T_{packet}$$
for one-channel

$$T_3 = (N_{RIOM}+2)\times(T_{tr}+T_{dec}+T_{under})+ N_{RIOM}\times(N_{DAC}+N_{DAT})\times T_{packet} T_{reject}$$
for two-channel

(1)

where,

N_{RIOM} : number of all RIOMs connected to the SIOM
N_{DAC} : number of DACs assigned to a RIOM = 1 clock
N_{DAT} : number of DATs a RIOM samples = 6 clocks
T_{packet} : time taken to transfer a packet = 3 clocks
T_{reject} : time to reject invalid data stream caused by switching channels. Its maximum is 20 clocks
T_{under} : time for tuned delays to avoid Receive FIFO underflow at a receiving station = 4 clocks.

Time parameters described in Fig. 7 are as follows.

T_{tr} Transmission time between each RIOM. Its average is 20 clocks at 20Mhz.

T_{dec} Time for each RIOM to decode its packet. This is 4 clocks at 20Mhz.

Hence, T_3 results in 273 clocks (13.65μs) for one-channel communication and 321 clocks (16.05μs) for two-channel communication, when N_{RIOM}=5.

T_4 takes up 2 clock cycles to send the last DAT and terminate the transaction at 20 Mhz.

T_5 is n+10 clocks for n 32-bit data across PCI bus in burst mode. In our case the entire 44 32-bit DAT data takes 54 clocks at 33 Mhz (1.62μs).

The total communication time is calculated to be T=16.49μs for one-channel communication and T=18.89μs for two-channel communication. Both cases satisfy our design requirement of 20 μs for communication latency.

B. Control loop time vs. Loss of system phase margin

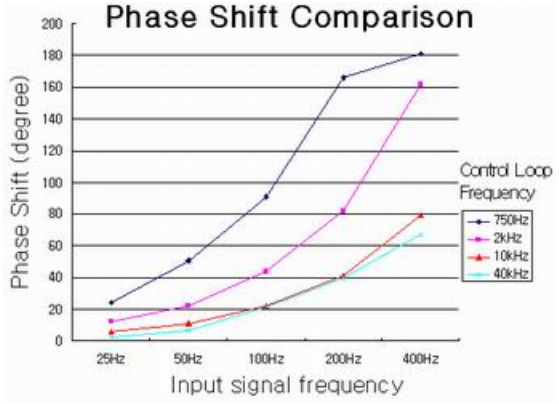

Fig. 9. Phase shift comparison vs input signal frequency and control loop frequency

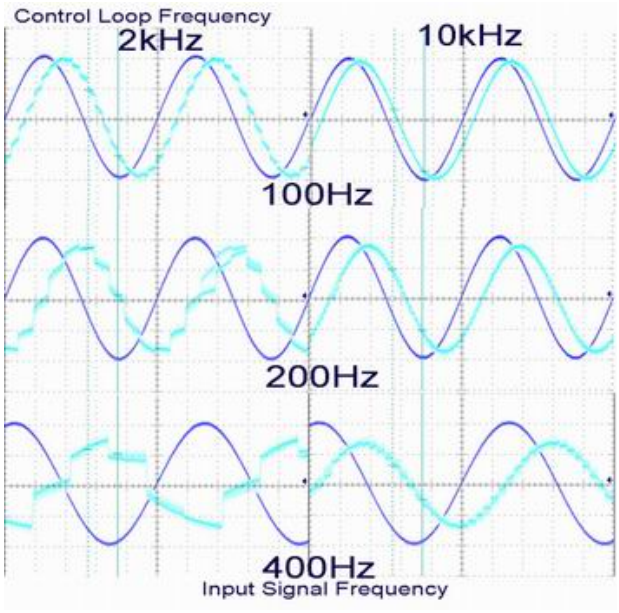

Fig. 10. Graphical results of Phase shift comparison

Fig. 9 and Fig. 10 illustrate the phase shift as a function of input signal frequencies at various control loop frequencies using one-channel communication.

Fig. 9 shows, for a given input signal frequency or desired system bandwidth, the faster the control loop frequency the smaller the influence of the latency on the phase shift. This ultimately dictate the stability and robust of the closed loop system.

Fig. 10 illustrates the effect of sampling and transmission delay on a sampled signals. The input frequencies at 100,

200 and 400 Hz were sampled at control loop frequencies of 2 kHz and 10 kHz.

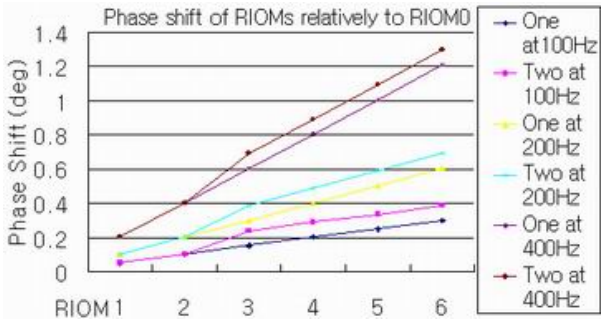

Fig. 11. Phase shift of RIOMs relatively to RIOM0

Fig. 11 presents the phase shift of RIOMs relatively to RIOM0 according to number of active channels and sampled signal frequencies. One and Two in the legend represents one-channel and two-channel communication, respectively, and the two-channel is the case when 3 RIOMs are on the first channel and 4 RIOMs on the second channel. Sampled signal frequencies are 100, 200 and 400Hz. The values are relative phase shift of RIOMs to RIOM0 which is the first RIOM on the first channel. Thus, if a RIOM is added on the ring, it has $0.05°$ penalty in its phase shift than its previous RIOM in order. And if it is added on the second channel, it has $0.086°$ penalty more.

VI. Future work

Future development is on going to incorporate the ExoPCI and the SIOMs boards into a single board. This will reduce power consumption as will as eliminate the time delay due to data transferred over the Parallel bus. The CY7C924ADX transceivers are replaced with CYP15G0201DXB and CYP15G0401DXB. The maximum transfer rate for new transceiver is up to 1.5Gbps, which is 7.5 times faster than the current transceiver.

VII. Conclusion

This paper has presented a high-speed real-time serial communication system and its protocol on an application for the Berkeley Lower Extremity Exoskeleton (BLEEX). The conventional control system utilizing a parallel interface for such a complex nonlinear MIMO systems would have resulted in a wiring nightmare of over 200 wires for the required sensors and actuators. Instead the high speed serial communication network dramatically reduce the wires to only 24.

The communication architecture was presented and how the ring topology was established. The communication goal was discussed and the properties of the HotLink transceiver for high-speed serial communication was described. As a result, the communication time for the control system measured, and found to satisfied the need for BLEEX to shadow its pilot while walking [8].

In conclusion, the drive for faster and faster internet communications has resulted in high speed serial communication chip sets which enables the use high-speed serial network for complex real-time control systems.

References

[1] J. Winkler and J. Munn, *Standars and Architecture for Token-ring Local Area Networks*, IEEE, 1986.

[2] E. Thomopoulos, L. E. Moser and P. M. Melliar-Smith, *Latency Analysis of the Totem Single-Ring Protocol*, IEEE, 2001.

[3] W. Seminarium and V. Universiteit, *Network Protocols*, Computing Surveys, Vol.13, No.4, Dec 1981

[4] A. Shionozaki and M. Tokoro, *Control Handling in Real-Time Communication Protocol*, 1993, ACM press, pp 149-159

[5] M. Törngren, *Fundamentals of Implementing Real-Time Control Applications in Distirubed Computer Systems*, Real-Time Systems, 14, 1998, pp 219-250

[6] CYPRESS CY7C924ADX Datasheet, *http://www.cypress.com/cfuploads/img/products/CY7C924ADX.pdf*

[7] CYPRESS application note, *Understanding Bit-Error-Rate with HOTLink*, *http://www.cypress.com/cfuploads/support/app_notes/ber.pdf*

[8] Berkeley Lower Extermity Exoskeleton Project website, *http://www.me.berkeley.edu/hel/bleex.htm*