Control System Development for a Novel Wire-Driven Hyper-Redundant Chain Robot, 3D-Trunk

KeJun Ning and Florentin Wörgötter

***Abstract*—This paper presents the control system for our novel hyper-redundant chain robot system "3D-Trunk" demonstrating an operational principle that is much different from traditional solutions. Its main features are that all the joints are passive, state controllable and share common inputs introduced by wire-driven control. For this unique design, a force-oriented method is employed to control the driving wires. The mechanical analysis, as well as an analysis of the differential driven mechanism of this design is formulated. The design of a novel wire tension state sensing component and its operation are also described. The system is controlled by distributed embedded controllers. The actuators' coordination mechanism and the bang–bang controller-based closed-loop control implementation of this novel prototype are discussed from a mechatronic system level. Thus, this paper, together with a predecessor [1], presents all required details allowing for building and controlling 3D-Trunk.**

***Index Terms*—Embedded system, hyper-redundant robot, robotics, wire-driven control.**

I. Introduction

A TRADITIONAL hyper-redundant chain robot (HRCR) is a serial manipulator that possesses more joints than the required degrees of freedom (DOFs). Generally, such structures achieve an increased level of dexterity [2], [3].

Constructing HRCR systems is difficult because tradeoffs between different restrictions have to be considered [4], concerning mechanical and electronic aspects. Being congruent with common open-chain schematics, the most popular approach to build an HRCR is by connecting several rigid links via an actuated revolute joint in a chain [1], [4]. Development of control system for such HRCR system is also an important concern. Generally, distributed control architecture and kinds of serial communication technologies are employed.

Manuscript received August 9, 2010; revised December 3, 2010 and March 17, 2011; accepted April 23, 2011. Recommended by Technical Editor M. O'Malley. This work was supported in part by PACO_PLUS from the European Commission, in part by the BMBF BFNT project 3a of the University Göttingen, and in part by the Bernstein Center Göttingen, project D1.

K. Ning was with the Bernstein Center for Computational Neuroscience, Institute of Physics III, University of Göttingen, 37077 Göttingen, Germany. He is now with Corporate Technology, Siemens, Ltd., Beijing 100102, China (e-mail: nkj@sjtu.org).

F. Wörgötter is with the Bernstein Center for Computational Neuroscience, Institute of Physics III, University of Göttingen, 37077 Göttingen, Germany (e-mail: worgott@physik3.gwdg.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2011.2151202

As compared to traditional mechanical actuation approaches, wire-driven (cable-driven and wire-actuated) systems offer an alternative by utilizing an actuated windlass to pull a wire for pulling an object or driving the joints of the machine. By this, wire-driven solutions can fulfill some special application requirements (compact size, light-weight, and easy to implement and maintain). But such systems also have certain drawbacks, e.g., limited positioning/driving accuracy and stiffness. In spite of this, their applications are nonetheless manifold, and we can use wire-driven mechanisms to create various compact and ingenious designs. Accordingly, several wire-driven robotic systems have been reported so far in the literature (e.g., [1], [5]–[26]), on parallel kinematics mechanisms (PKMs) [9]–[19], HRCRs [1], [7], [8], [20]–[23], robotic hands (e.g., [24]–[27]), and general nonredundant serial manipulators [28].

Depending on the working principle of a wire-driven robotic system, different control solutions are employed. For wire-driven PKMs, wires have been utilized instead of rigid links and/or joints, and the end-effector (movable platform) is controlled by the length of each wire [9], [15]. For wire-driven serial robots, actuated wires are used to drive the joints like tendons or muscles. Different from using wires to drive rigid links of a robotic system (hand, manipulator, etc.), the work presented in [7], [8], and [28] uses wires to drive many elastic links for some special purposes. Since thin wires are easy to route, such a serial robot may have more DOFs contained in a compact configuration. Furthermore, several interesting and original HRCR designs exist [5]–[8], which offer new concepts for driving long chains by fewer actuators thereby limiting the weight and size of the systems.

These original wire-driven designs and solutions [1], [7], [8], [20]–[23] are all trying to implement complex chain robots by overcoming size and weight restrictions. At the same time, however, manipulation abilities or dexterousness might be partially sacrificed or restricted. This is due to the fact that some degree of coupling exists in these designs for sharing the input power [7], [8], [22].

In [1] and [5], we presented a novel concept for building an HRCR, and the implemented prototype system, 3D-Trunk, was used to introduce a new design paradigm. Its basic mechanical design issues and the computational model (kinematics and dynamics) are described in [1]. Different to the design shown in [7], [8], [22], and [28], all joints of 3D-Trunk are fully decoupled and independently controllable [1].

In this paper, we formulate this novel design's wire-driven principle and focus on the control system development for

the original platform disclosing all corresponding technical solutions.

The remainder of this paper is organized as follows. In Section II, an overview of the design is given. The concept and some key components of 3D-Trunk are shown. The principle of a novel wire state sensing mechanism is also provided. Section III describes the closed-loop wire-driven control of this new HRCR. A bang–bang controller is employed to control the driving wires. Subsequently, the system-level design and wire-driven considerations as well as the solutions are discussed. In Section IV, we disclose the key issues on the distributed embedded control systems development for 3D-Trunk. Experimental results of this prototype are provided in Section V. Finally, conclusions are presented in Section VI. In the Appendix, Section VII, we analyze and formalize the underlying mechanisms and provide details of the differential driven principle of this design. The control issues discussed in the main parts of the text are based on this.

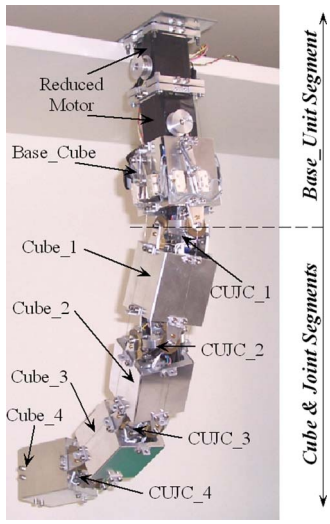


Fig. 1. Original prototype, 3D-Trunk is an 8-DOF wire-driven system. All electronic components and microcontrollers are embedded inside. CUJC is the joint with state controllability. Cubes are used for positioning these CUJCs.

II. Overview of 3D-Trunk's Design

One motivation of this paper is to present novel design concepts/solutions to construct a complicated motion machine owning many DOFs with few actuators. In this section, we give an overview of the novel HRCR's mechanical design. This background information is the foundation for the control system development, described afterward.

The core concept of this new design is that all joints of the HRCR are passive and state controllable and share common inputs introduced by wire-driven control, but they have to work in an asynchronous mode [1]. Based on this concept, the implementation is highly modular and scalable no matter how many DOF are implemented. Fig. 1 shows our original prototype system 3D-Trunk, which consists of a "Base_Unit" and many identical modularized "Cube & Joint" segments. At present, 3D-Trunk is an 8-DOF wire-driven system, powered by four motors.

Fig. 2(a) shows 3D-Trunk's driving chain design. The Base_Unit is the key segment of the HRCR. As shown in Fig. 2, on one end, it connects to the first cube by a controllable universal joint component (CUJC). On the other end, four reduced motors are fixed for actuating the whole chain. There are four wire tension state sensing components (WTSSCs) symmetrically mounted on the four outer faces of the Base_Unit. The WTSSC is important to this system's wire-driven control (see later).

For 3D-Trunk, the CUJC introduces joint state controllability; and at any one moment in time, only a single unlocked joint is differentially driven by a pair of driving wires. Due to the unidirectional force from the wires, we need a pair of driving wires to actuate one joint, similar to muscles and tendons. As the joints are perpendicularly arranged in an alternating way (see Fig. 2), two pairs of driving wires are enough to drive the system.

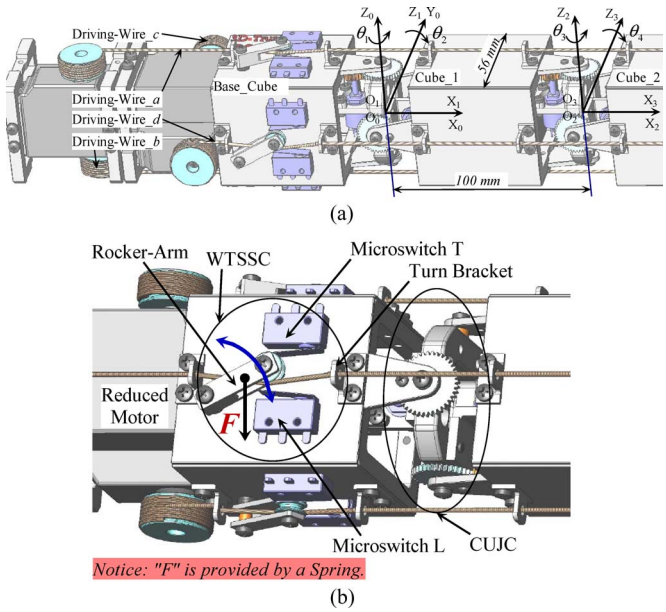


Fig. 2. 3D-Trunk's mechanical design. This wire-driven system has state controllable passive DOFs and shared driving, no matter how many DOF are implemented. By unlocking any one joint, its joint angle can be actuated by a pair of driving wires. (a) Driving chain design, as well as the Denavit–Hartenberg (D–H) coordinate frames design. (b) Base_Cube, for housing four reduced motors and a four-way WTSSC. The WTSSC is the key component to sense the driving wires' states.

A. 3D-Trunk's Controllable Joint Design

In this paragraph, we provide information on 3D-Trunk's unique joint design, CUJC. Each CUJC provides two independent joint angle measurement and state control mechanisms. As shown in Fig. 3, this compact design is based on a Hooke joint (2-DOF). Here, a linear solenoid (pull type) driving mechanism was employed to construct a compact binary-state clutch for one DOF [1] and [5]. The moveable iron core of the solenoid (see Solenoid_Actor in Fig. 3) is used to lock and unlock its

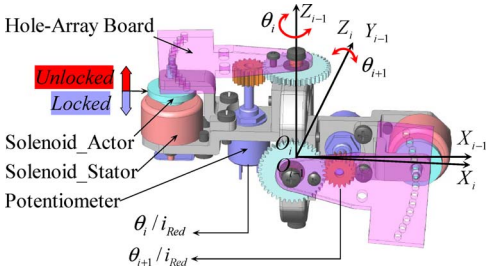


Fig. 3. CUJC design for 3D-Trunk. One CUJC offers two independent and state-controllable DOFs, with joint-angle measurement. This is another core mechanical component of this novel system.

joint via a hole-array board (connected with the cube), which provides arrayed holes to plug into. A potentiometer is used to measure a joint's absolute angle. A pair of gears is employed to transmit motion and increase measurement resolution. The angle feedback is not only used to measure the joint's real rotation to achieve closed-loop control but also to check the result of a locking (plugging) action (see later).

B. 3D-Trunk's Wire State Sensing Implementation

A central component of any wire-driven system is the tension sensor needed to control the tension of the wires. This will be introduced next and compared to other solutions. In [17], strain gauges were mounted to an "E-"shaped frame for measuring and deducing wire tension. A similar design is also discussed in [29]. For a strain-gauge-based method, we have to assemble and calibrate the sensors very carefully, and this solution is not cheap and not good for measuring weak tension. Our solution is new and different from this, with some specific characteristics and a special operation method.

The principle of the WTSSC is explained in Fig. 4(a). For a WTSSC, a rotatable rocker arm is pulled by a spring and triggers a pair of microswitches. If the wire is too loose, the spring will push the rocker arm to press the loose-side switch, whereas the tense-side switch will be depressed and vice versa.

Fig. 4(b) shows the relation between the wire's pulling force and the rocker arm's rotation angle β. Let us denote the angles where the two states (loose and tense) are triggered as β_L and β_T, respectively. Table I explains this design. Given that these microswitches are pulled up by resistors, zero means that the switch is pressed. The WTSSC is a discrete design, so in region 1 of Fig. 4(b), the wire is in an unmeasurable state. Once the wire is too loose, it may hop out from its windlass. This is a critical problem for this wire-driven system, due to the fact that wires can only be pulled.

In spite of the fact that this WTSSC is only a three-state discrete solution, we can also obtain a derivative, which provides useful additional information, by a special operating method. As shown in Fig. 4(b), if we control the wire to "drive" the WTSSC to alternate between "1 0" and "1 1" rapidly, all the time, then the tension of the wire will be close to F_{NL}. The threshold F_{NL} needs a special driving method (dither-type tension control), and it acts as a pretight force, which is important for the control of our HRCR as will be discussed later.

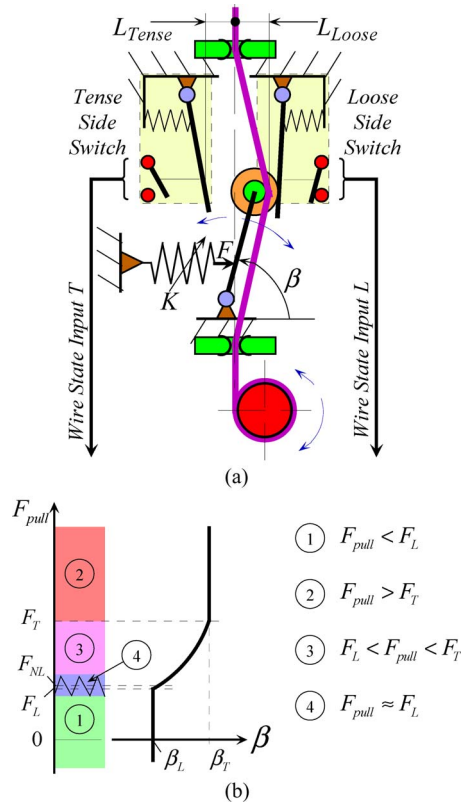


Fig. 4. WTSSC design for 3D-Trunk. (a) Principle of the WTSSC. If the wire is too loose, the spring will push the rocker arm to press the loose-side switch, whereas the tense-side switch will be depressed and vice versa. This is a discrete solution for deducing the pulling force of a wire with resilience capability. (b) Relation between the wire pulling force and the rocker-arm's rotation angle. By a dither-type tension control, we can obtain a useful additional state, the region 4. For regions 1–4, see Table I.

TABLE I
EXPLANATION OF THE WTSSC OUTPUT

Region	WTSSC_Output (T L)	Pressed Microswitch	Driving-Wire's state
1	1 0	L	Loose, in an un-measurable state.
2	0 1	T	Tense, can provide strong enough pulling force.
3	1 1	None	Not-loose and not-tense.
4	Alternating between "1 0" and "1 1"	L is pressed rapidly and intermittently.	The tension of the wire is close to F_{NL} [see Fig. 4 (b)]. F_{NL} is the pulling force's mean value, defined by the "duty cycle".

By adjusting the related parameters shown in Fig. 4(b), e.g., L_{Loose} and L_{Tense}, the spring's elastic coefficient K, etc., we can deduce and obtain suitable state-threshold definitions. Thus, this way, we obtain a discrete solution for deducing the pulling force of a wire with resilience capability.

In Fig. 4(a), if a rotation sensor was employed for measuring the rocker arm's rotation angle β, we could obtain more accurate results, but would need more controller resources [e.g., analog/digital converter (ADC) port]. Instead, the solution presented in Fig. 4(a) is simple and very practical.

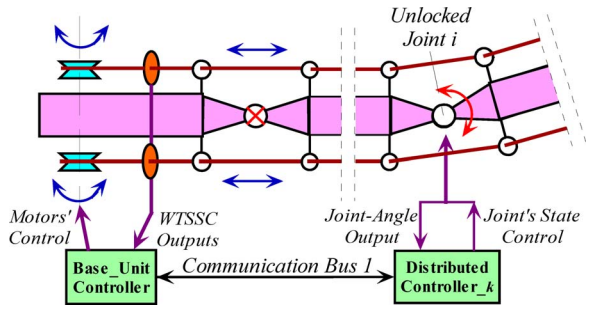


Fig. 5. Closed-loop wire-driven control principle of 3D-Trunk. A distributed control architecture is employed for this system. To complete closed-loop position control, the Base_Unit Controller needs to keep communicating with the distributed controller for getting the current position of the controlled joint. Using this it controls the motors to actuate the effective pair of driving wires.

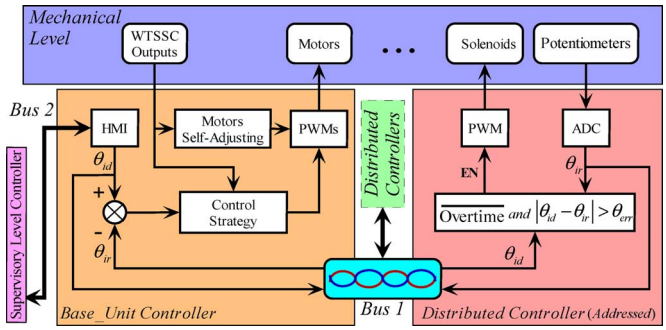


Fig. 6. Implementation schematics of closed-loop wire-driven control of 3D-Trunk. The functional partitions of the embedded controllers and the control flow between the components are shown. The Base_Unit Controller receives commands from an HMI, communicates with the distributed controllers, and coordinates the motors to drive the HRCR. The distributed controllers control the joints' states and acquire the joints' angles in real time.

Up to now, we presented the key mechanical component design of 3D-Trunk. This information is the basis for the following control solutions.

III. Closed-Loop Wire-Driven Control Principle of the Joints

The previous section has introduced the key components of 3D-Trunk. In this section, we further discuss how to operate these components and give solutions for controlling the system.

For 3D-Trunk, a distributed control architecture is employed. Thus, to achieve coordinated movements of the HRCR, one needs the distributed controllers to cooperate. As shown in Fig. 5, to complete closed-loop position control, the Base_Unit Controller needs to keep communicating with the distributed controllers. This way, it gets the current position of the controlled joint and controls the motors to actuate the effective pair of driving wires, concurrently.

Please note, for the schematics shown in Fig. 5, calculating the driving-wires' length changes for controlling windlass rotation is highly unpractical, as some special issues have to be addressed. For example, the geometrical situations, which occur during this HRCR's spatial motion, bring wires' length/tension changes and need to be compensated. Such a calculation would be further burdened by the wire's elasticity and the distributed frictions involved, as well as by the fact that the wrapped wire leads to an effective variation of the diameter of the windlass, etc. Therefore, for 3D-Trunk, we use a force-oriented solution, which is based on the aspects described in Section II and differential driven principle provided in the Appendix, Section VII.

Furthermore, 3D-Trunk is relies on the concept of a "DOF controllable and driving shared" solution [1], which is different from the traditional joint driving ways. Thus, we need to control a joint's state, and at the same time, coordinate the driving wires to achieve closed-loop control at joint level, depicted by the conceptual overview shown in Fig. 5. The whole sequence of operation steps to drive one joint can be summarized as follows:

1) unlocking the joint that will be moved;
2) applying the wire-driven control strategy (bang–bang control, see later) on a pair of wires to pull the selected mobile joint toward its target position;
3) applying a tension maintaining method (dither-type control, see later) to maintain the tensions in the remaining wires;
4) locking the movable joint and stopping all motors, when the moving joint is within a tolerance of the target position.

A more detailed implementation schematics is shown in Fig. 6, which provides more information on the embedded controllers' functional partitions and the control flow between the components mentioned earlier. Here, we need to give more explanations about the implementation of the tension-maintaining method and the employed wire-driven control strategy. The former is used to drive the noneffective pair; and the latter is in charge of finishing the closed-loop wire-driven control at joint level.

As mentioned in Section II-B, essentially, the WTSSC supports force-oriented control. The fourth state shown in Table I [region 4, also in Fig. 4(b)] is an approach to obtain a semiconstant force (F_{NL}) by a dither-type tension control on the discrete WTSSC design. Based on (9) (see Section VII, Appendix), this F_{NL} can be used to drive the negative side and the noneffective pair wires. This is the easiest way to ensure the stability of this wire-driven robot (overcoming the coupling problems of this nonlinear system and the wire's length change caused by its elasticity) and to reduce the power dissipation. On an embedded platform, we can implement this dither-type tension controller easily, as shown in Fig. 7. In fact, this self-triggered routine is robust against disturbances and consumes only limited resources.

For 3D-Trunk, a bang–bang control-based solution is employed to construct the control strategy shown in Fig. 6, which steers the effective pair wires to achieve positional closed-loop control of an unlocked joint. The bang–bang mechanism is a minimum time optimization feedback controller that switches abruptly between two states and is robust against perturbations. For this solution, there are several considerations that we have taken into account. First, since the joints have discrete resolution

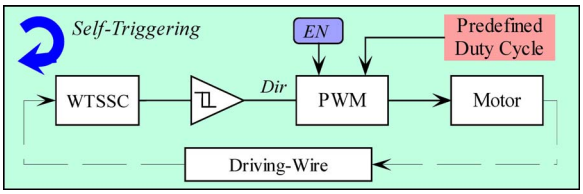


Fig. 7. Self-triggered routine ensures that the driving wire maintains a weak pretight state F_{NL}. Implementing this dither-type control on a microcontroller is computationally cheap and efficient. By adjusting the duty cycle, we can change F_{NL} to some degree [see Fig. 4(b)].

(see the hole-array board in Fig. 3), we can naturally and directly define this as the error band parameter θ_{err}, which serves as the hysteresis of the bang–bang controller. Second, as shown in Fig. 4 and Table I, once the WTSSC enters region 2 of Fig. 4(b), this component will not limit the motor to provide strong enough driving force for the unlocked joint. This means that we can impose a strong enough force on the positive side wire. Third, the bang–bang mechanism is easy to implement on a microcontroller. These aspects improve the feasibility of our design.

However, different from the traditional single-input and single-output bang–bang controller, for the movable joint of 3D-Trunk, the effective pair wires need to be driven synchronously. Our solution, thus, combines ideas from finite-state machine with the traditional bang–bang controller, shown in Fig. 8. In this way, we can efficiently and transparently model our new control strategy and assure all requirements of the wire-driven characteristics of this HRCR.

As shown in Fig. 8(a), we use the driving wire's "role" (the role can be "positive-side wire," "negative-side wire," or "non-effective pair") as the input to the controllers of the motors. By checking the real-time relation between $|\theta_{id} - \theta_{ir}|$ and θ_{err}, all wire roles are determined (see Section VII, Appendix), and then we can control the corresponding driving wires. Here, θ_{id} is the given target joint angle, and θ_{ir} is this joint's present position, which can be obtained by accessing the corresponding potentiometer (see Fig. 3).

Thus, as shown in Fig. 8(b), the input is the "role," and the results F_k, $k = a, b, c, d$, are approximately proportional to the pulsewidth modulation (PWM) duty cycle for the four motors. Please note to achieve the conditions listed in Fig. 8(b), the dither-type controller discussed earlier has been employed for getting F_{NL}. Thus, during driving a joint, there are three concurrent routines (shown in Fig. 7) required for accessing and operating the WTSSCs.

So far, we have disclosed the key control components running on the Base_Unit Controller shown in Fig. 5. And now, we can go back to Fig. 6 and take a look at the control flow. In fact, Fig. 6 is the basis for the firmware development of all the distributed microcontrollers.

As shown in Fig. 6, once 3D-Trunk receives a motor command by the human–machine interface (HMI), it will understand it and operate accordingly. For example, if joint *i* is required to go to a desired angle θ_{id}, the system will finish the closed-loop control by itself. To complete this task, the Base_Unit Controller needs to tell the distributed controllers to unlock joint *i* and get the

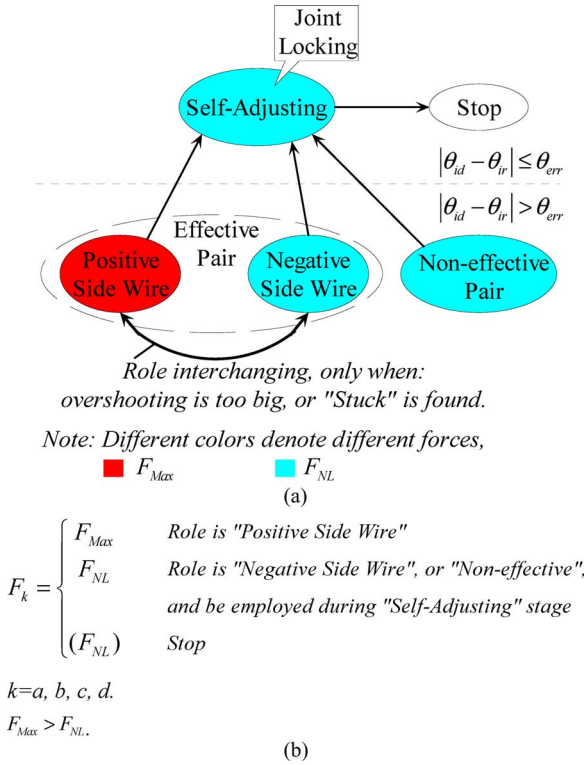


Fig. 8. Bang–bang control-based solution for driving the driving wires. Here, we use the driving wire's "role" (i.e., "positive-side wire," "negative-side wire," or "noneffective pair," see Appendix) as the input of the motors' controller. This figure shows the core of the control strategy in Fig. 6. (a) Control state mechanism design for actuating the driving wires. All the wires' roles are depending on the real-time relation between $|\theta_{id} - \theta_{ir}|$ and θ_{err}. (b) Control policy for the motors. Because the role of a driving wire is decided from (a), we can control the corresponding motor's PWM signal to adjust the driving tension.

real-time joint angle θ_{ir} of joint *i*, then dynamically actuate the driving wires according to the previous control strategy. These dynamic processes will be repeated and the information has to be updated in real time. Once the Base_Unit Controller finds the updated joint angle within an acceptable position interval θ_{err}, it will send a command to the distributed controller to lock joint *i* immediately.

Fig. 9(a) further exhibits the coordination and control of these actuators. The period from t_1 to t_2 is used for waiting for the unlocking action. During t_2 to t_3, the positive-side wire is controlled to drive the joint to the desired position, and from t_3 to t_4, all wires are self-adjusted to a pretight state [region 4 in Fig. 4(b), and Table I].

Up to now, we have presented the concept of this novel HRCR and disclosed all low-level control related components, as well as the principle of closed-loop wire-driven control. The practical implementation on the embedded controllers is discussed next.

IV. Embedded Control Systems of 3D-Trunk

The following section focuses on some technical details, which are essential for building 3D-Trunk. Readers, who are only interested in the general concept can probably skip this section. Hence, we first describe the general hardware control

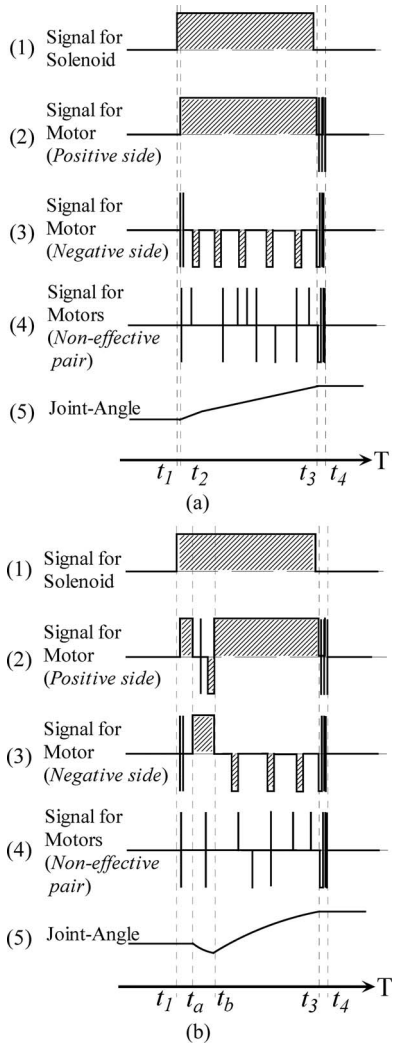


Fig. 9. Explanation of the actuators' (motors and addressed solenoid) coordination and driving. Note, the amplitude of the signal does not correspond to the duty cycle of the PWM signal; it just represents the active operating state. (a) Coordination and driving signal design for wire-driven. (b) Coordination and driving design for dealing with exceptional situations (see Appendix).

architecture of 3D-Trunk, which is modular and fully embedded into the mechanical housings. This is followed by some technical details on the communication and the individual control components.

A. Hardware Control Architecture of 3D-Trunk

The block diagram in Fig. 10 shows a detailed overview of the hardware architecture of 3D-Trunk's control system. It is an open and extendable solution. From an engineering perspective, to build such an HRCR, it clearly makes sense to distribute sensing and actuation electronics (PWM generation and actuation, ADCs, and the basic processing operations).

The Base_Unit Controller (see Figs. 6 and 10) consists of three small printed circuit boards (PCBs), all encased in the Base_Cube. Two of them provide interfacing circuits and four-way H-bridge driving circuits for the four dc reduced motors. The third one is the microcontroller board. The distributed controllers are encased in Cube_1 and Cube_3 (see Fig. 1). Each entire distributed controller consists of two small PCBs. One is the microcontroller board, and the other one is the MOSFET-Array board. The microcontroller boards share the same PCB layout. Using multiple smaller separated circuit boards makes it easy to encase the entire control system into the compact space of the cubes, and to avoid interference between the movable mechanical parts and the inside mounted PCBs.

The employed microcontroller is an ATMEGA16 (AVR core, from ATMEL), running at 16 MHz. All the in-system-programming [30] ports of the microcontrollers are externally accessible (from outside of 3D-Trunk), in order to facilitate the firmware development and debugging.

B. Communication Bus

This section describes the communication development for 3D-Trunk. The communication between the Base_Unit Controller and the distributed controllers is via an RS-485 serial bus, because of its high immunity to noise and its ability to drive large distances with high data rates, leading to the opportunity to design very long such HRCRs.

As shown in Fig. 10, the Base_Unit Controller and the distributed controllers are all independent nodes of the Communication Bus 1. Because the RS-485 is a half-duplex bus, avoiding transfer conflicts has to be considered in the related firmware code. For 3D-Trunk, the Base_Unit Controller is the master node and controls the transmission direction of the data.

As the ATMEGA16 has a single hardware universal asynchronous receiver transmitter (UART) to send and receive bytes, an extended, software-driven UART was independently implemented in the Base_Unit Controller.

A command set was implemented for this system. All distributed controllers receive command packets broadcasted by the Base_Unit Controller and parse them. Each command packet contains an ID and a unique operation code, so only the addressed distributed controller (ID being matched) will respond and behave according to the operation code.

At present, the implemented command set includes motion control, the joints' present position acquisition, the joints' zero-position ADC value acquisition, the system running state acquisition, some test and running demos, etc. More functions can be included. Based on this set of commands, an operator can easily control the system to complete some work by application program on a PC.

C. Distributed Controllers

Finally, we need to describe the controllers for the Base_Unit and the distributed controllers of the CUJCs, starting with the latter.

Each distributed controller has a unique ID and contains the controlled CUJCs' private information (e.g., the supervised joints' zero-position ADC values, some calibration values, etc.). This makes the system better modularized and repairable.

As shown in Fig. 3, small plastic conductive potentiometers were employed in our prototype, for sensing the joints' absolute angles with low electrical noise, high linearity, and long life.

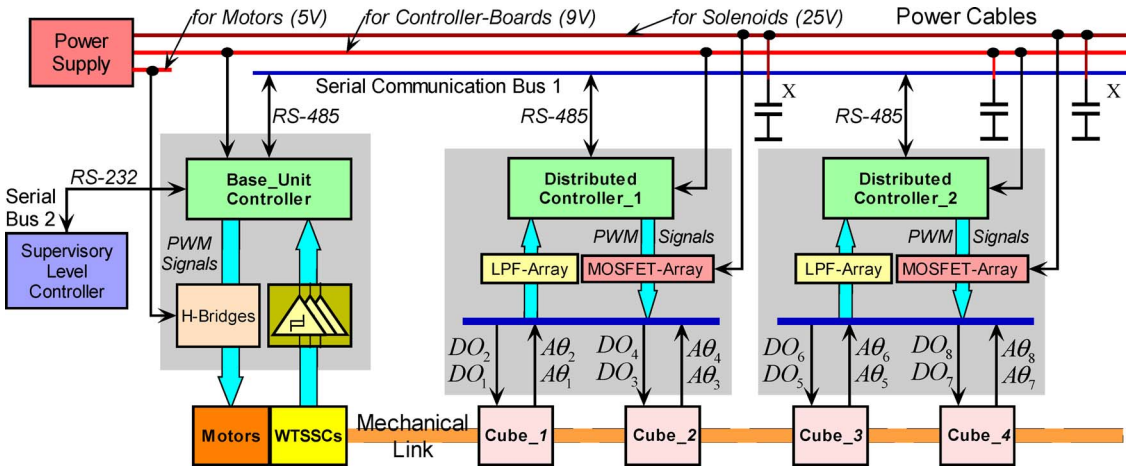


Fig. 10. Distributed control system hardware architecture of 3D-Trunk. The core electronics components of the Base_Unit Controller and the distributed controllers are shown. The power cables consist of three branches, for actuating electronic circuits, motors, and solenoids, respectively. These controllers and all cables are embedded inside 3D-Trunk's cubes.

These potentiometers were connected with the microcontroller's ADC pins, via simple *RC* low-pass filter circuits. The peripheral of Timer2 [30] runs at 100 Hz; its interrupt service routine (ISR) periodically triggers the controller's ADC routine for updating the joint-angles' information.

The n-channel power MOSFET transistors, mounted on the MOSFET-Array board, switch the currents of the CUJCs' solenoids. The distributed controller switches these MOSFETs by generating PWM signals. Timer0 runs at 10 KHz to generate a time base for the PWM, and all solenoids share this signal with a state index. The solenoid's overtime protection mechanism is implemented in the distributed controllers, for avoiding possible overheating [see Fig. 6]. The intention to power these solenoids by PWM current is to obtain much higher pulling force while the solenoid will still not burn out.

All data transferred on the Communication Bus 1 (see Fig. 10) will be received and analyzed by all distributed controllers. Once one of the distributed controllers finds a valid command packet; thus, the contained ID is under its supervision; it will execute this command immediately. The other distributed controllers will ignore this packet and do nothing.

D. Base_Unit Controller

The Base_Unit Controller is in charge of controlling the four dc reduced motors by interfacing the microswitches of the WTSSCs (via Schmitt triggers) to deduce the driving wires' tension states. As shown in Fig. 10, the Base_Unit Controller also acts as the bridge between Serial Bus 1 and 2. The command interpreter and the system's coordination mechanism are also implemented in the Base_Unit Controller's firmware. For the development of the firmware, we borrowed some ideas from operating systems to construct the system's coordination mechanism.

The Base_Unit Controller needs to drive four dc motors concurrently, in the course of adjusting the driving wires' initial tensions and for the joint's closed-loop control. The detailed control and coordination mechanism design have been disclosed in Section III. Here, the related driver routines of the firmware share a Timer2 peripheral of the microcontroller with several index and state variables.

For the communication happening on Serial Bus 2 (see Fig. 10), the Base_Unit Controller's UART ISR always pushes the received message into a queue pool, and its main routine will pop up and process this information. Once a valid command packet is found, it will start the data transfer with the distributed controllers on the Communication Bus 1 and drive the Base_Unit Segment to operate the Cube & Joint Segment (shown in Fig. 1). The Base_Unit Controller is the key controller of the system, because it not only needs to concurrently operate the motors and WTSSCs, but it also coordinates the distributed controllers' work and controls the whole system's operation (see Fig. 6).

Some other assistant functions were also implemented in the Base_Unit Controller's firmware. For example, under some conditions, a Solenoid_Actor may fail to be pulled out from the hole-array board by the magnetic force generated by the powered Solenoid_Stator. This can happen when the torque applied to the corresponding joint is too big and the magnetic force is not strong enough to move the "stuck" Solenoid_Actor (see Fig. 3). If such an exceptional situation happens, the driving wires will fail to drive this joint. The Base_Unit Controller will recognize this problem by receiving an unchanging joint angle. Then, it will reverse the current role of the effective pair of wires instantly for a short time to counteract the external torque and to help releasing the stuck Solenoid_Actor. The related control signals are shown in Fig. 9(b). At the same time, the Base_Unit Controller keeps checking this joint's angle. Once the Controller finds the angle changing (hence, the joint is unstuck), it will resume its former driving action to finish the task. In Fig. 9(b), this course is from t_a to t_b. Tests on the prototype with and without this heuristic problem-solving mechanism have proven its usefulness. Such kinds of embedded autonomous functions improve the whole system's working reliability and stability.

So far, we described the key issues and solutions for the embedded control systems development of 3D-Trunk showing

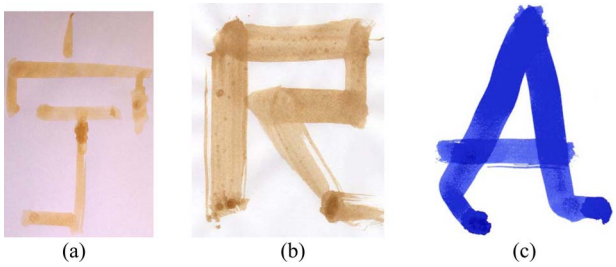


Fig. 11. Characters finished by 3D-Trunk. (a) Chinese character "Ning," one meaning is silence. (b) and (c) English characters: "R" and "A."

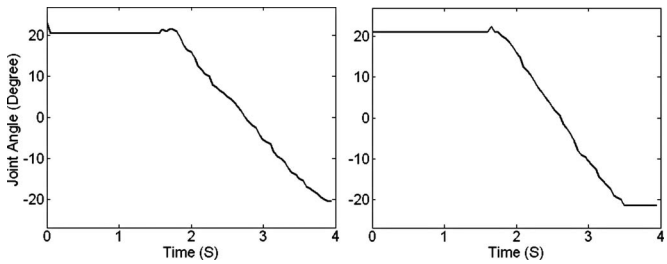


Fig. 12. Real-time response curves with autonomous escaping from a "stuck" situation (Joints 1 and 3). Sampling frequency is 20 Hz and the sampling time is 4 s.

that the system level design of 3D-Trunk offers a high potential for extensions.

V. Experiments

Following design and construction, the performance of 3D-Trunk was verified by several different experiments. A movie demonstrating its feasibility and maneuverability is shown in [31]. For a certain set of experiments, we extended the design by an additional 1-DOF end-effector (paint brush holder) [5]. This platform was then employed to do several painting experiments. Fig. 11(a) shows a Chinese character completed by 3D-Trunk, and the corresponding movie is shown in [32]. Furthermore, 3D-Trunk is also employed to conduct monocular vision guided operations [33].

Please note, limited by the concept of this design, the joints sharing the same effective pair of driving wires can only be driven asynchronously. In Section VII, Appendix, we discussed the 2-DOF simultaneous driving mode. We can use it to obtain twice the movement speed in a shape changing process, or achieve a 2-D coordinated motion. For instance, for painting work, we can achieve diagonal strokes by driving an odd and an even numbered joint at the same time. Fig. 11(b) and (c) shows two English characters, "R" and "A," finished by 3D-Trunk, using this method. For the character "A," this result exhibits the "crosstalk" effect. Generally, this crosstalk will only influence the transition but will not effect the final locking positions of the two simultaneous driven joints. For the design shown in Fig. 13(c), this crosstalk will not happen.

In Section V, we discussed the solution for possible "stuck" situations occurring between the Solenoid_Actor and the hole-array board (see Fig. 3). Real-time response curves for this solution are shown in Fig. 12. At the beginning, the Solenoid_Actor

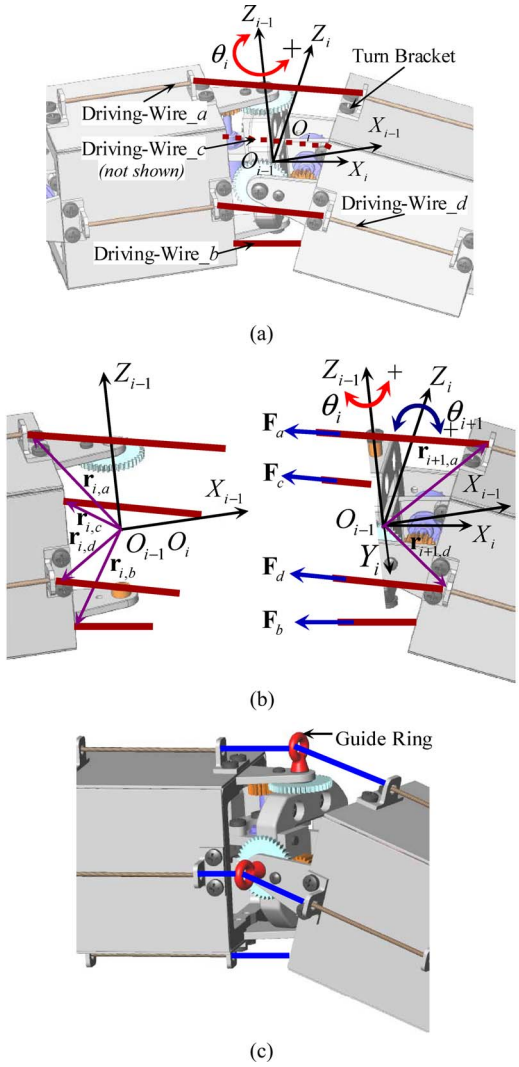


Fig. 13. Two pairs of wires drive this system. For one joint, only one pair of wires is effective. (a) For the unlocked joint i, wires c and d are the effective pair. (b) Driving wires' effect applied on the unlocked joint i. The rotation axis is Z_{i-1}. For any one joint j, just replace the subscript i with j. (c) By adding guide rings to restrict the driving wires, respectively, the possible "crosstalk" during a simultaneous driving mode will be eliminated.

failed to be pulled out from the hole-array board by the magnetic force generated by the powered Solenoid_Stator. The Base_Unit Controller detected this problem and behaved following the coordination mechanism design introduced in Fig. 9(b) and Section IV to help freeing the stuck Solenoid_Actor. The small rising peak (around 1.8 s in the two cases shown in Fig. 12) is mostly the result from the backlash between the Solenoid_Actor and hole-array board. By this backlash, the method described in Fig. 9(b) is made to work. The response curves are recorded from the movement sequence shown in [31].

VI. Conclusion

In this paper, we presented the driving principle and control system development for our HRCR design, 3D-Trunk.

This paper builds on earlier publications [1], [5], [33] and describes its novel wire-driven principle and control system

development. The first two studies [1] and [5] had solely described the novel concept, the original mechanical design giving a short overview of the distributed architecture and robotics basis. The third one [33] equips 3D-Trunk with monocular vision to do a vision-guided experiment. The implemented inverse kinematics of this redundant robot is also disclosed.

In this paper, the coordination and control principle are now fully described, with many detailed design considerations and solutions. This way, the documentation of 3D-Trunk is now complete allowing others to reproduce design and control.

The current control is not based on calculating the wires' changing lengths during motion, but it is a force-oriented method. This well-founded method guaranteed the functionality of our implementation. The distributed joint-angle sensors and the WTSSCs design have provided suitable solutions for implementing the method presented in this paper.

In this paper, the implemented control is based on bang–bang control, to achieve a minimum-time solution. Being equipped with low-power motors, as well as discrete joint angles, this solution is suitable. On the other hand, transitions within a movement sequence are slightly disturbed during start and stop of the motion. This is one limitation of the bang–bang control technology. Proportional control could be employed in a future version to alleviate this problem.

Contrasting our design against the prototypes presented in [7], [8], [23], and [28], 3D-Trunk is a complete closed-loop system design. As by design [1], all joints of our HRCR are fully decoupled, independently controllable and measurable and have very few actuators.

Furthermore, the present system level design of 3D-Trunk is highly modularized and scalable, both, concerning mechanical as well as electronic aspects. It is, thus, a totally novel design and quite different in many aspects compared to traditional solutions.

3D-Trunk works in an asynchronous mode and has discrete resolution, and it cannot implement a multiple joints concurrent motion. These aspects limit its applications. But, it is quite suitable to be employed for many shape-changing applications, particularly with high holding torque requirements (the solenoid clutches lead to very high torque resistance, see [1]). As a consequence, we believe our work is helpful for others to develop affordable HRCRs, as well as for stimulating new inventions and improvements for constructing novel mechanisms and/or mechatronic systems for some special applications: for example, not only to construct an HRCR, but also to build a multi-DOF shape changing system with only few actuators.

Appendix: Differential Driven Principle of This Design

In this appendix, we derive the equations resulting from the differential driven principle of our approach. This serves as the proof of operation for our control solutions.

For an *N*-DOF HRCR, the $N\times 1$ input torque vector associated with the generalized coordinates is as follows (see Fig. 2 and [1]):

$$\mathbf{T} = [T_1, T_2, \ldots, T_N]^T. \tag{1}$$

Here, we need to model the relation between the input torque vector $\mathbf{T}$ and the actuation arising from the driving wires and the locking mechanisms.

An analysis scenario is shown in Fig. 13. As the joint is a 1-DOF revolute type, we only need to discuss the projected component on the rotation axis (for joint *i*, the rotation axis is $O_{i-1}Z_{i-1}$). The "driving" torque of any one joint is as follows:

$$T_j = T_{Wj} + T_{Lj}, \qquad j = 1, 2, \ldots, N \tag{2}$$

where T_{Lj} is the torque generated by the routed locking mechanism (see Figs. 2 and 3), associated with the rotation axis $O_{j-1}Z_{j-1}$; T_{Wj} is the composite torque (component on axis $O_{j-1}Z_{j-1}$) generated by the routed four-way driving wires [see Fig. 13(b)].

If joint *i* is unlocked, then for all joints of this HRCR

$$\begin{cases} T_{Lj} = 0 & j = i \\ T_{Lj} \neq 0 & j \neq i. \end{cases} \tag{3}$$

Then,

$$T_{Wi} = T_i, \qquad i \,\text{is the unlocked joint} \tag{4}$$

and

$$T_{Lj} = T_j - T_{Wj}, \qquad j = 1, 2, \ldots, N \quad \text{and} \quad j \neq i. \tag{5}$$

Generally, for an arbitrary pose

$$T_{Wi} \neq T_{Wj}, \qquad i, j = 1, 2, \ldots, N \quad \text{and} \quad i \neq j. \tag{6}$$

Equation (2) shows that essentially each joint of this HRCR is driven by five actuation inputs in parallel, which is much more complicated than a traditional robotic joint design.

If the input torque vector can be obtained (e.g., by inverse dynamics calculation [1]), (4) means that the unlocked joint's torque, generated by the driving wires, should be equal to T_i for obtaining the desired performance. For any one locked joint, its "clutch" provides the remaining required torque to satisfy (2). Equation (5) shows how to calculate the clutch torques of the locked joints. This is similar to utilizing an actuator to keep a fixed angle of this joint (i.e., balancing the dynamic and gravity loads).

Refer to the case shown in Fig. 13(b), for any one joint *j*, T_{Wj} in (2) is as follows:

$$T_{Wj} = \mathbf{Z}_{j-1} \bullet (\mathbf{r}_{j,a} \times \mathbf{F}_{j,a} + \mathbf{r}_{j,b} \times \mathbf{F}_{j,b} + \mathbf{r}_{j,c} \times \mathbf{F}_{j,c} + \mathbf{r}_{j,d} \times \mathbf{F}_{j,d}), \qquad j = 1, 2, \ldots, N \tag{7}$$

where $\bullet$ and $\times$ mean dot product and cross product, respectively; $\mathbf{Z}_{j-1} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ is the unit vector of axis $O_{j-1}Z_{j-1}$; $\mathbf{F}_{j,a} \sim \mathbf{F}_{j,d}$ are the pulling forces along the routed driving wires *a*--*d*; $\mathbf{r}_{j,a} \sim \mathbf{r}_{j,d}$ are the vectors from O_{j-1} to the guiding centers of the distributed turn brackets [for positioning and guiding the driving wires, see Fig. 2(b)]. The values for $\mathbf{r}_{j,a} \sim \mathbf{r}_{j,d}$, as well as the orientations of $\mathbf{F}_{j,a} \sim \mathbf{F}_{j,d}$ can be obtained from the mechanical design (see Fig. 13); they are depending on the vector of joint variables $\mathbf{q} = [\theta_1, \theta_2, ..., \theta_N]^T$ and described in the same coordinate frame.

If we ignored the friction between the turn brackets and the driving wire, then the tension force along a wire would be

constant. Let us define $F_k = |\mathbf{F}_k|$, $k = a, b, c, d$, to describe the driving force generated by the four reduced motors [see Figs. 1 and 2(a)].

In the case shown in Fig. 13(b), for the unlocked joint $j = i$, we have $T_{Li} = 0$. Then, the four driving wires' contributions can be grouped to $T_{i,ab}$ and $T_{i,cd}$ as follows:

$$\begin{cases} T_{i,ab} = \mathbf{Z}_{i-1} \bullet (\mathbf{r}_{i,a} \times \mathbf{F}_{i,a} + \mathbf{r}_{i,b} \times \mathbf{F}_{i,b}) \\ T_{i,cd} = \mathbf{Z}_{i-1} \bullet (\mathbf{r}_{i,c} \times \mathbf{F}_{i,c} + \mathbf{r}_{i,d} \times \mathbf{F}_{i,d}) \end{cases}. \tag{8}$$

Equation (8) exhibits that the torque generated by the wires *a* and *b* are mostly perpendicular to the axis of joint *i* [see Figs. 2 and 13(b)], and thus have limited effect on joint *i*. If we reduce them (i.e., F_a and F_b) based on some considerations, then this pair of wires' composite effect will be so small that we can totally ignore them.

Referring to Fig. 13(a) and (8), if there is no external load and $\mathbf{Z}_{i-1} \bullet (\mathbf{r}_{i,c} \times \mathbf{F}_{i,c} + \mathbf{r}_{i,d} \times \mathbf{F}_{i,d}) > 0$ (e.g., winding wire *c* and unwinding wire *d*), joint *i* (θ_i) will rotate positively and vice versa. For the remaining locked joints, the distributed locking mechanisms will provide torques to satisfy (5), and thus, the wires' traction will not affect the locked joints. So, we define here for the unlocked joint *i* that wires *c* and *d* form the "effective pair," and wires *a* and *b* the "noneffective pair."

So that based on (4), (7), and (8) and the earlier discussion, for the one currently unlocked joint *i* and the coordinate frame design shown in Fig. 2(a), we can arrive at

$$\begin{aligned} T_i &= T_{i\_\mathrm{Eff}\_\mathrm{Pair}} \\ &= \begin{cases} \mathbf{Z}_{i-1} \bullet (\mathbf{r}_{i,c} \times \mathbf{F}_{i,c} + \mathbf{r}_{i,d} \times \mathbf{F}_{i,d}) & \text{if } (i = 1, 3, 5, \ldots) \\ \mathbf{Z}_{i-1} \bullet (\mathbf{r}_{i,a} \times \mathbf{F}_{i,a} + \mathbf{r}_{i,b} \times \mathbf{F}_{i,b}) & \text{if } (i = 2, 4, 6, \ldots) \end{cases}. \end{aligned} \tag{9}$$

And we can conclude that every unlocked joint of this HRCR is differentially driven by a pair of wires (i.e., the effective pair).

As shown in (9), for the case in Fig. 13(a) to achieve a positive rotation of joint *i*, wire *c* is wound tighter for a positive contribution, while the tension in wire *d* (F_d) makes a negative contribution. For easing to the following discussion, we call the wire with positive contribution (reducing the error toward the target angle) the "positive-side wire" and the other one of the effective pair the "negative-side wire."

Please note, there are always two inputs in (9) and they should be always positive (wire can only be pulled). For our implementation, we assign F_{NL} for "noneffective pair" and "negative-side wire," see Fig. 8.

Essentially, (9) presents the relation between the effective pair's pulling forces and the unlocked joint's driving torque. Combining (4), (5), and (9) and the dynamics model introduced in [1], we can conduct exact dynamics analysis, perform part checking [i.e., verifying the design of the locking mechanism (clutch), checking the specifications of the solenoid, choosing motors, etc.], as well as performance pursued control.

Furthermore, (9) discloses the basis to achieve a 2-DOF simultaneous driving mode: we can drive an odd and an even numbered joint at the same time (see painting experiment), but "crosstalk" between the wire pairs could occur. By symmetrically mounting guide rings [see Fig. 13(c)] to restrict the driving wires, respectively, such crosstalk during simultaneous driving mode will be eliminated.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers who have made many valuable comments, which improved this paper.

REFERENCES

[1] K. Ning and F. Wörgötter, "A novel concept for building a hyper-redundant chain robot," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1237–1248, Dec. 2009.

[2] S. Chiaverini, G. Oriolo, and I. D. Walker, "Kinematically redundant manipulators," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Germany: Springer, 2008, ch. 11, pp. 245–268.

[3] G. Robinson and J. B. C. Davies, "Continuum robots—A state of the art," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, 1999, pp. 2849–2854.

[4] E. Shammas, A. Wolf, and H. Choset, "Three degrees-of-freedom joint for spatial hyper-redundant robots," *Mech. Mach. Theory*, vol. 41, pp. 170–190, Feb. 2006.

[5] K. Ning and F. Wörgötter, "A DOF state controllable & driving shared solution for building a hyper-redundant chain robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, 2009, pp. 5880–5885.

[6] A. Ananiev, I. Kalaykov, E. Petrov, and B. Hadjiyski, "Single-motor driven construction of hyper-redundant robot," in *Proc. IEEE/ASME Int. Conf. Mechatronics Robot.*, Aachen, Germany, 2004, pp. 549–553.

[7] K. L. Paap, M. Dehlwisch, and B. Klassen, "GMD-Snake: A semi-autonomous snake-like robot," presented at the 3rd Int. Symp. Distributed Autonomous Robotic Systems, Saitama, Japan, 1996.

[8] I. D. Walker and M. W. Hannan, "A novel 'elephant's trunk' robot," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Atlanta, GA, 1999, pp. 410–415.

[9] H. Kino, S. Yabe, C. C. Cheah, S. Kawamura, and S. Arimoto, "A motion control scheme in task oriented coordinates and its robustness for parallel wire driven systems," in *Proc. 9th Int. Conf. Adv. Robot.*, Tokyo, Japan, 1999, pp. 545–550.

[10] S. Kawamura, W. Choe, S. Tanaka, and S. R. Pandian, "Development of an ultrahigh speed robot FALCON using wire drive system," in *Proc. IEEE Int. Conf. Robot. Autom.*, Piscataway, NJ, 1995, pp. 215–220.

[11] K. Maeda, S. Tadokoro, T. Takamori, M. Hiller, and R. Verhoeven, "On design of a redundant wire-driven parallel robot WARP manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, 1999, vol. 2, pp. 895–900.

[12] J. S. Albus, R. Bostelman, and N. G. Dagalakis, "The NIST ROBOCRANE," *J. Robot. Syst.*, vol. 10, no. 5, pp. 709–724, 1993.

[13] T. Maier and C. Woernle, "Flatness-based control of underconstrained cable suspension manipulator," in *Proc. DETC'99 ASME Design Eng. Tech. Conf.*, Las Vegas, NV, 1999, pp. 1–10.

[14] S. Kawamura, H. Kino, and C. Won, "High-speed manipulation by using parallel wire-driven robots," *Robotica*, vol. 18, pp. 13–21, 2000.

[15] K. Ning, M. Zhao, and J. Liu, "A new wire-driven three degree-of-freedom parallel manipulator," *ASME Trans. J. Manuf. Sci. Eng.*, vol. 128, pp. 816–819, Aug. 2006.

[16] L. Notash and A. Kamalzadeh, "Inverse dynamics of wire-actuated parallel manipulators with a constraining linkage," *Mech. Mach. Theory*, vol. 42, pp. 1103–1118, Sep. 2007.

[17] Y. Mae, T. Arai, and K. Inoue, "Tension control and force estimation of hybrid drive parallel arm," *J. Robot. Soc. Japan*, vol. 21, no. 4, pp. 436–443, 2003.

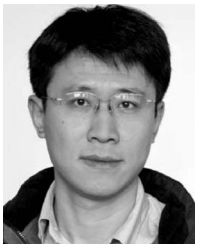
[18] S. Tadokoro, Y. Murao, M. Hiller, R. Murata, H. Kohkawa, and T. Matsushima, "A motion base with 6-DOF by parallel cable drive architecture," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 2, pp. 115–123, Jun. 2002.

[19] P. H. Borgstrom, N. P. Borgstrom, M. J. Stealey, B. Jordan, G. S. Sukhatme, M. A. Batalin, and W. J. Kaiser, "Design and implementation of NIMS3D, a 3-D cabled robot for actuated sensing applications," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 2, pp. 325–339, Apr. 2009.

[20] R. Buckingham, "Snake arm robots," *Ind. Robot: An Int. J.*, vol. 29, no. 3, pp. 242–245, 2002.

[21] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *J. Robot. Syst.*, vol. 20, no. 2, pp. 45–63, 2003.

[22] R. Cieslak and A. Morecki, "Elephant trunk type elastic manipulator—A tool for bulk and liquid materials transportation," *Robotica*, vol. 17, pp. 11–16, Jan. 1999.
[23] A. Kapoor, N. Simaan, and R. H. TayJor, "Suturing in confined spaces: Constrained motion control of a hybrid 8-DoF robot," in *Proc. 12th Int. Conf. Adv. Robot.*, Seattle, WA, 2005, pp. 452–459.
[24] I. Yamano and T. Maeno, "Five-fingered robot hand using ultrasonic motors and elastic elements," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 2684–2689.
[25] Y. Ogahara, Y. Kawato, K. Takemura, and T. Maeno, "A wire-driven miniature five fingered robot hand using elastic elements as joints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, 2003, pp. 2672–2677.
[26] N. Imamura, M. Kaneko, and T. Tsuji, "Development of three-fingered robot hand with a new design concept," in *Proc. IASTED Int. Conf. Robot. Manuf.*, Banff, Canada, 1998, pp. 44–49.
[27] M. Kaneko, M. Higashimori, R. Takenaka, A. Namiki, and M. Ishikawa, "The 100 G Capturing Robot – Too Fast to See," *IEEE/ASME Trans. Mechatronics*, vol. 8, no. 1, pp. 37–44, Mar. 2003.
[28] J. Peirs, H. V. Brussel, D. Reynaerts, and G. D. Gersem, "A flexible distal tip with two degrees of freedom for enhanced dexterity in endoscopic robot surgery," in *Proc. MME, The 13th Micromechanics Eur. Workshop*, Sinaia, Romania, 2002, pp. 271–274.
[29] J. S. Sulzer, M. A. Peshkin, and J. L. Patton, "Pulling your strings," *IEEE Robot. Autom. Mag.*, vol. 15, no. 3, pp. 70–78, Sep. 2008.
[30] Atmel Corporation. (2005, Jun.). ATmega16(L). Rev. 2466L-06/05 [Online]. Available: www.atmel.com
[31] K. Ning and F. Wörgötter. (2009, Oct.). "3D-Trunk, A novel concept hyper-redundant chain robot (2)," [Online]. Available: http://www.youtube.com/watch?v=oVTMS4VdYUs
[32] K. Ning and F. Wörgötter. (2009, Oct.). "A novel concept hyper-redundant chain robot: 3D-Trunk," [Online]. Available: http://www.youtube.com/watch?v=C1PuO9cwmj0
[33] K. Ning and F. Wörgötter, "To paint what is seen: A system implementation of a novel conceptual hyper-redundant chain robot with monocular vision," in *Proc. ISR/ROBOTIK (41st Int. Symp. Robot.)*, Munich, Germany, 2010, pp. 722–727.



KeJun Ning received the B.S. (Hons.) and M.S. (Hons.) degrees in mechanical engineering from Northeastern University, Shenyang, China, in 1999 and 2002, respectively, and the Ph.D. degree in mechatronics engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006.

From May 2006 to July 2007, he was a Senior Mechatronic System Engineer at Shanghai Grandar Robotics Company., Ltd. From August 2007 to May 2008, he was a Senior Engineer and Researcher in the Robotics Department, China Corporate Research Center, Shanghai Branch, ABB (China) Ltd., China. From May 2008 to September 2010, he was a Postdoctoral Researcher at the Bernstein Center for Computational Neuroscience, University of Göttingen, Göttingen, Germany. He is currently a Lead Research Scientist at Corporate Technology, Siemens, Ltd., Beijing, China. His research interests include the areas of robotics, mechanism design, embedded systems, wireless sensor networks, and autonomous systems.

Florentin Wörgötter studied biology and mathematics at the University of Düsseldorf, Düsseldorf, Germany. He received the Ph.D. degree focused on the visual cortex from the University of Essen, Essen, Germany, in 1988.

From 1988 to 1990, he was engaged in computational studies at the California Institute of Technology, Pasadena. Between 1990 and 2000, he was a Researcher at the University of Bochum, Bochum, Germany, where he was investigating the experimental and computational neuroscience of the visual system. From 2000 to 2005, he was a Professor of computational neuroscience in the Psychology Department, University of Stirling, Stirling, U.K., where his interests strongly turned toward "Learning in Neurons." Since July 2005, he has been the Head of the Computational Neuroscience Department, Bernstein Center for Computational Neuroscience, Institute of Physics III, University of Göttingen, Göttingen, Germany. His current research interests include information processing in closed-loop perception-action systems, sensory processing, motor control, and learning/plasticity, which are tested in different robotic implementations. His group has also developed the RunBot, which is a fast and adaptive biped-walking robot.