

An Embedded Controller for a Quadruped Robot

Xiaoqi Li, Wei Wang and Jianqiang Yi, Senior Member, IEEE

Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

{xiaoqi.li, wei.wang, jianqiang.yi}@ia.ac.cn

Abstract - The design of controller is of critical importance for capability of the quadruped robot. In this paper, we present an embedded controller scheme, with ARM and DSP microprocessors, to control the locomotion of a quadruped robot. The controller consists of an ARM board for planning and a DSP board for motion control. Regarding the sensor information, the ARM board is responsible for making plans about switching among gaits, speeding up and slowing down, and communicates with the DSP board via serial parallel interface (SPI). The DSP board, containing four DSP processors, takes charge of receiving ARM command, calculating to generate the motor command, and communicating with DC motor drivers via CAN bus. Finally, we implement the designed controller on a quadruped robot to control the movement of a limb. Test result demonstrates the adequacy of the controller to execute accurate and real-time control.

Index Terms - *Embedded controller, ARM, DSP, Serial parallel interface (SPI), CAN.*

I. INTRODUCTION

Quadruped robots have received much attention recently for their ability to traverse unstructured environments robustly and efficiently. Concerning these advantages, a variety of quadruped robots which possess different characteristics have been developed to execute dynamic tasks such as search and rescue. The quadruped robot BigDog is able to perform many locomotion behaviors, such as crawl, trot, bound and gallop gait with rough-terrain mobility [1]. LittleDog is a small four-legged robot designed for research on legged locomotion [2], and has realized statically walking, and dynamically bounding on rough terrain [3-4]. MIT Cheetah is a highly efficient legged robot which can achieve dynamic running gait [5]. HyQ is designed to fulfill highly dynamic tasks like jumping and running [6-8]. An ape-like robot Charlie is developed to increase the efficiency of the walking robot by using intelligent structures, and has the ability to stand up, walk, and balance on the slope [9] [10].

Although many quadruped robots as mentioned above have realized static and dynamic locomotion, the control of quadruped robots is still considered to be difficult and complex for the strong coupled and nonlinear characteristics of the quadruped robot. Therefore, the controller is predominant for the capability of the quadruped robot.

BigDog's behavior is controlled by an onboard computer, which manages the sensors, records engineering data, and handles communications with a remote human operator [1]. LittleDog robot is controlled by an onboard x86, 266 MHz CPU with 128 MB RAM and a Mini-PCI 802.11 wireless Ethernet interface [2]. HyQ's all joints are controlled by an onboard Pentium computer connected to motor and valve

drivers [6]. Tekken3 utilizes a vision PC for navigation, and the walking speed and direction are sent to the servo controller DIMM-PC [11]. The MIT Cheetah utilizes i7 dual-core 1.33 GHz CPU as the Real-Time controller [12].

The controllers for robots mentioned above all adopt commercial-off-the-shelf products, and generally satisfy the requirement. However, considering the user customized interfaces, low-power consumption, and small size, some researchers select to develop custom designed embedded controller.

The ape-like robot Charlie handles the ankle joint control, the spine control as well as the communication tasks using FPGA-electronic boards [10, 13]. Mobile robots "EyeBot family" all carry a small and versatile embedded controller which combines a 32bit Motorola M68332 with a number of standard interfaces and drivers for DC motors, servos, several types of sensors, and a digital color camera [14].

We are developing a quadruped robot QR-I, as depicted in Fig. 1. The torso of QR-I is 0.4 m in length, 0.2 m in width, and the height of QR-I is 0.33 m. The total weight of QR-I is 20 kg. Each limb of QR-I has a hip pitch joint and a knee pitch joint, and each joint is actuated by a DC motor.



Fig. 1 The on-going developed quadruped robot QR-I at our lab.

Considering the interfaces, size, weight, and power consumption, the commercial-off-the-shelf controller is not suitable for QR-I. So we present an embedded controller scheme for the quadruped robot QR-I.

The embedded controller is made up of an ARM board and a DSP board. The ARM board communicates with the DSP board via SPI, and the DSP board communicates with two motor drivers via CAN bus. The controller mainly accomplishes the following functions: sensor signal modulation and acquisition, planning according to the sensor information, calculation to get the motor control command and control the motor drivers to actuate DC motors to realize

locomotion of the quadruped robot QR-I.

In the remainder of the paper, we firstly give an overview of the system architecture. Following this, we detail the SPI communication, CAN communication, and foot-ground-reaction force sensing. Afterwards, we utilize the controller on QR-I, and show the test result. Finally, we draw the conclusions and describe the future work.

II. SYSTEM ARCHITECTURE OVERVIEW

As illustrated in Fig. 2, the overview of the system architecture can roughly be divided into two parts: (1) the embedded controller which contains an ARM board and a DSP board, and (2) the limb unit. The limb unit exists for each limb of the quadruped robot, but is not repeated in the figure.

The ARM board utilizes an ARM processor as the core, integrated with LCD display interface, Ethernet interface, USB keyboard interface, RS232 serial communication interface (SCI), SPI interface, and the sensor signal modulation circuit, as shown in Fig. 2 and Fig. 3. SamSung's S3C6410 (ARM11) [15] is used in the ARM board for its basic frequency is 667 MHz at 1.2 V, which is suitable to realize real-time control. The ARM processor can communicate with one or several external PCs via Ethernet bus and RS232 interface, and exchange data with the four DSPs via SPI interface. Based on the sensor information, the ARM processor makes plans about which gait the quadruped

robot adopts and the locomotion speed, and sends the planning command to the DSP board.

The DSP board (shown in Fig. 2 and Fig. 3) uses four TI's TMS320F2812 (DSPs) as core processors which possess a high executing speed of 150MIPS [16-17] to perform complex algorithms in real time with high precision, and integrated with JTAG interface, SPI interface and CAN interface. Each of the four DSPs controls a hip joint motor driver and a knee joint motor driver in one limb of the quadruped robot. Based on the received ARM command, the four DSPs calculate to get the desired motor position value, and transfer the control command to motor drivers via CAN bus.

Each of the four limb units consists of a foot sensor, two motor drivers and two DC motors, as depicted in Fig. 2 and Fig. 3. Each motor driver actuates one motor using arbitrarily positioning, speed and current regulation. The two motors are responsible for the motion of hip joint and knee joint of one limb of the quadruped robot, respectively. Four foot sensors are mounted under the four feet of the quadruped robot, respectively.

An external power supply module provides 24VDC to motor drivers. In addition, the 24V external power supply is transferred to 5 V by XR10/24S05 for sensor signal modulation circuit, 3.3 V and 1.8 V by TPS767D318 for ARM and DSP processors.

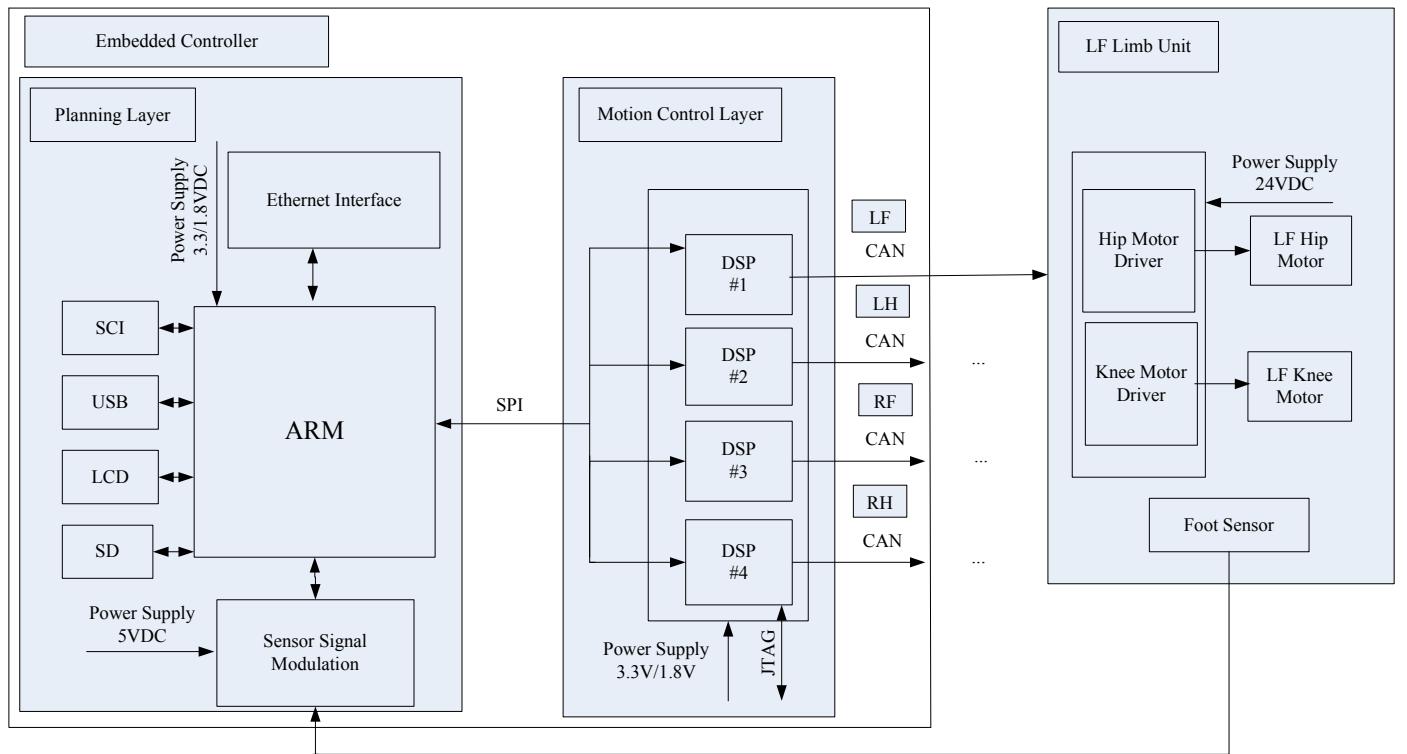


Fig. 2 Overview of the system architecture. We just show one limb unit here. The abbreviations LF, LH, RF and RH represent left fore, left hind, right fore, and right hind limbs, respectively.

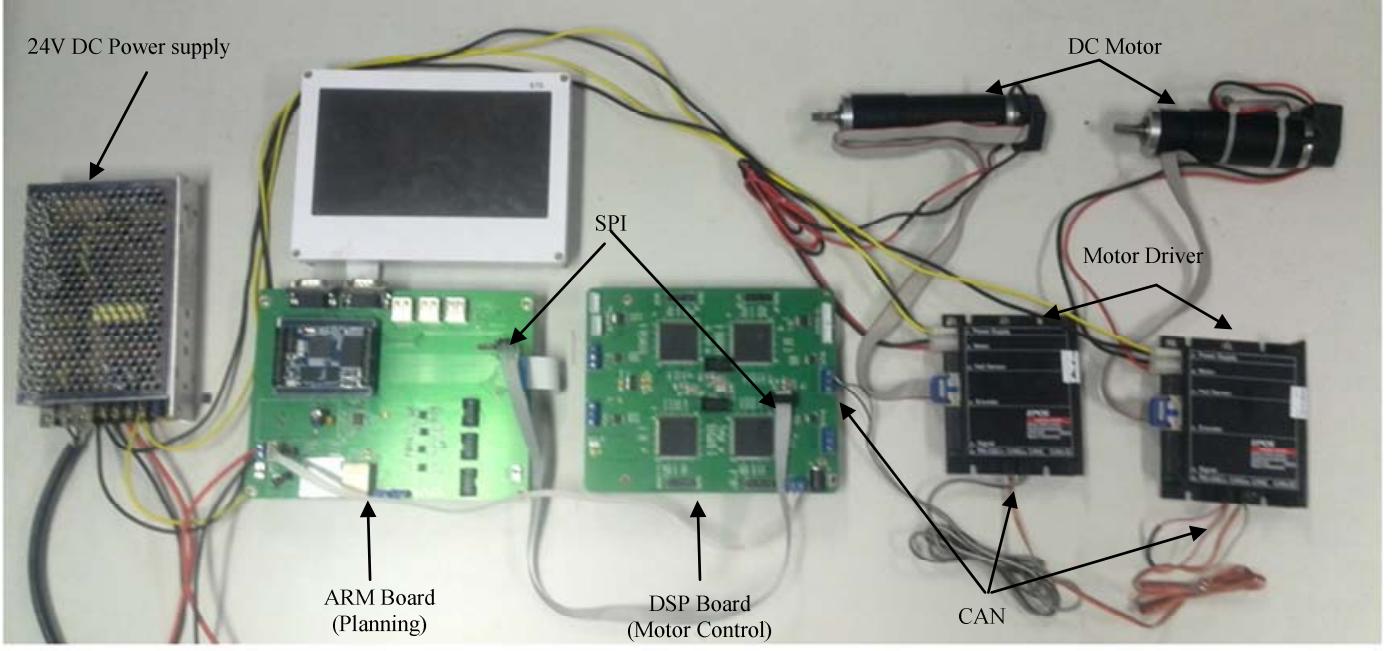


Fig. 3 Overview of the hardware of the system developed for the quadruped robot QR-I (Only two drivers and DC motors are shown here).

III. COMMUNICATION AND FOOT-GROUND-REACTION FORCE SENSING

Since the designed controller is distributed, it is a key point to harmonize the ARM, DSPs and the motor drivers via communication. SPI has advantages in signal integrity and supporting multiple slaves, so SPI is utilized to realize communication between ARM and DSP processors. CAN is a multi-master serial bus supporting two or more nodes. Therefore, we design that one DSP communicates with two motor drivers via CAN bus.

The designed controller is a closed loop one because it plans using the sensor information as feedback quantity. We select force sensing resistor (FSR) as the force sensor for its small size and high sensitivity. The FSR exhibits a change in resistance with different applied force. However, the ARM processor is not able to read resistance value directly. Therefore, the sensor signal modulation circuit is required to accomplish force-to-voltage conversion.

A. SPI Communication between ARM and DSPs

SPI is a duplex, synchronous serial communication interface used for short distance communication, primarily in embedded systems [18]. The full-duplex capability makes SPI simple and efficient for applications. So it is widely used to send data between microcontrollers and talk to a variety of peripherals, such as sensors and shift registers.

We utilize an ARM processor as the master, and four DSP processors as slaves in SPI communication, as shown in Fig. 4. The SPI clock is provided by ARM, and the four DSP processors share the MISO and MOSI lines. Four I/O pins of the ARM are used to provide independent CS signals for the four DSP processors.

To begin communication, the ARM uses a frequency supported by the slave DSP to configure the clock. Then the ARM sets the I/O pin to a logic level 0 to select the slave DSP, and keeps the rest three I/O pins high. During each SPI clock cycle, a full duplex data transmission occurs. The ARM sends a bit on the MOSI line and the selected DSP reads it, while the selected DSP sends a bit on the MISO line and the ARM reads it. For example, the ARM processor sends a byte of data 0x3e to DSP#1, and DSP#1 sends 0x4b to the ARM. We use the hardware shown in Fig. 3 to execute this SPI communication, and the detection signals of CS pin, CLK pin MOSI pin, and MISO pin are shown in Fig. 5. We can see that the ARM and DSP#1 are set to capture data on the clock's rising edge and propagate data on the clock's falling edge. In addition, the CS pin keeps logic low during data transmission.

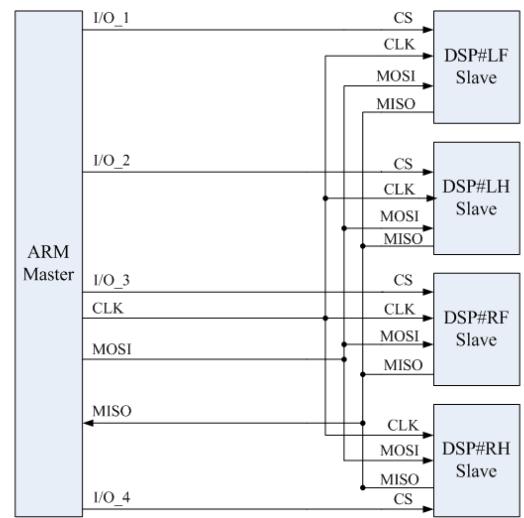


Fig. 4 Physical connection diagram of SPI between ARM and four DSPs.

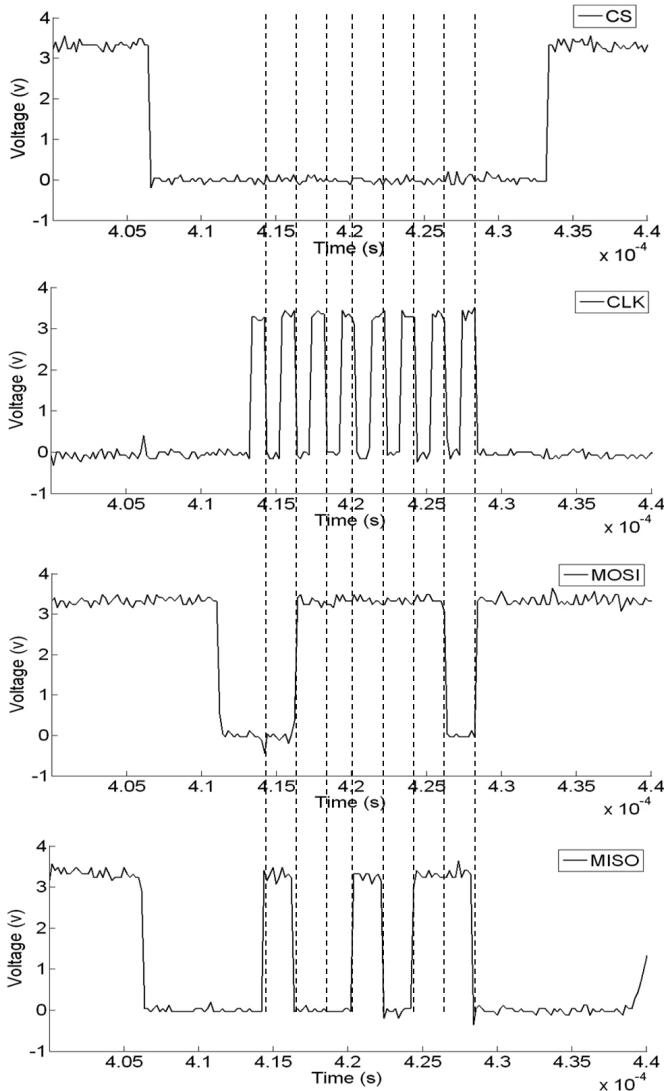


Fig. 5 Signals of CS, CLK, MOSI, and MISO pins detected by an oscilloscope.

B. CAN Communication between DSP and Motor Drivers

CAN is a kind of embedded network with transmission speed up to 1Mbit/sec. CAN bus provides a standard communication interface among different subsystems and hardware. Due to the advantages of high reliability and low cost protocol, CAN is used extensively in automotive and automation industries [19-20].

Each of the four DSPs are designed to communicate with a hip motor driver and a knee motor driver via CAN bus, as shown in Fig. 6. It is obviously shown that an advantage of CAN is to reduce the complexity of the wires and unnecessary connections. All nodes are connected to each other through a two wire bus, CAN_H and CAN_L, and each node is able to send and receive messages.

We utilize CANopen protocol to realize fast communication between a DSP and two motor drivers. Data are transferred as “frame” in CAN communication. When the DSP propagates the motor control command, 11 bit identifier

is transmitted at the start of the CAN frame. The motor driver receives the motor control command if its identifier is same with that in the CAN frame. Each motor driver on the network can independently transfer and receive messages. For example, we use the hardware shown in Fig. 3 to execute CAN communication between DSP#1 and two motor drivers. The detection signals of CAN_H and CAN_L pins are shown in Fig. 7.

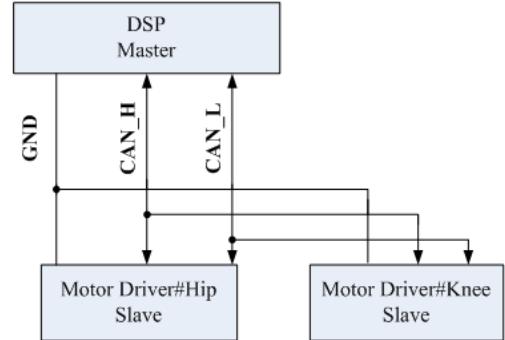


Fig. 6 Physical connection diagram of CAN interface between a DSP and two motor drivers.

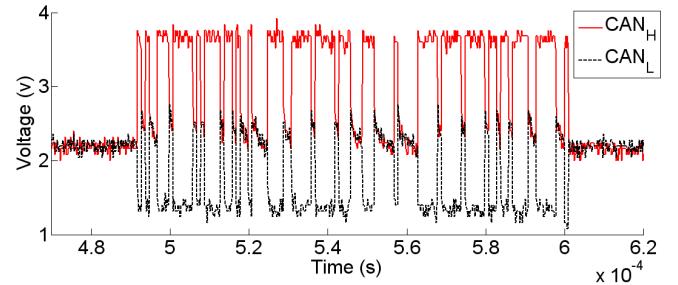


Fig. 7 Signals of CAN_H and CAN_L pins detected by an oscilloscope.

C. Foot-ground-reaction Force Sensing

To detect if the foot of the quadruped robot touches down the ground, we use force sensing resistor (FSR) which is a polymer thick film (PTF) device with small size, as shown in Fig. 8 (a). FSR exhibits a decrease in resistance with an increase in the force applied to the active surface, as shown in Fig. 8 (b).

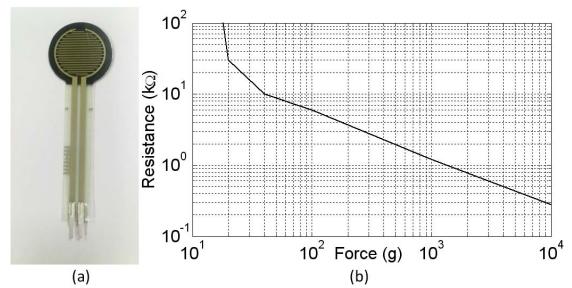


Fig. 8 (a) FSR force sensor. (b) Force- resistance characteristic curve.

For a force-to-voltage conversion, we design a non-inverting comparator as the sensor signal modulation circuit. The comparator with hysteresis uses op-amp LM358, as illustrated in Fig. 9 (a). The non-inverting input of the op-amp

is driven by the output of the divider (made up of R_1 and R_f), which is a voltage that increases with force. The resistor R_3 sets the hysteresis level. Hysteresis sets a lower and upper force threshold F_{f1} and F_{f2} to eliminate the multiple transitions caused by noise, as depicted in Fig. 9 (b). The x-axis represents the GRF applied to the FSR sensor, and the y-axis is the output of the comparator.

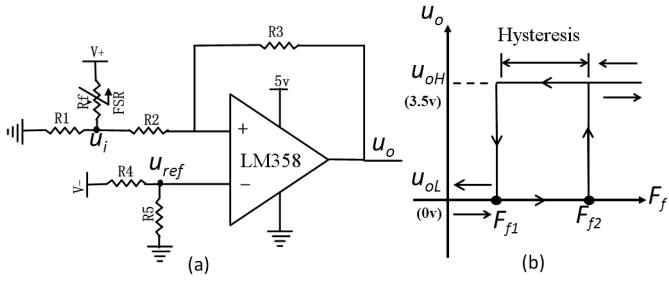


Fig. 9 (a) The sensor signal modulation circuit. (b) The output of the comparator with hysteresis.

The FSR is arranged in a voltage divider with R_1 , so we get:

$$U_i = \frac{R_1 V_+}{R_1 + R_f} \quad (1)$$

R_4 and R_5 are used as a voltage divider to drive the inverting input of the op-amp:

$$U_{ref} = \frac{R_5 V_-}{R_4 + R_5}. \quad (2)$$

The threshold voltages U_H and U_L are calculated by the following equations:

$$U_H = \frac{R_2}{R_2 + R_3} U_{ref} + \frac{R_3}{R_2 + R_3} U_{OH}. \quad (3)$$

$$U_L = \frac{R_2}{R_2 + R_3} U_{ref} + \frac{R_3}{R_2 + R_3} U_{OL}. \quad (4)$$

When the foot of the quadruped robot is off the ground and the limb is in swing phase, the resistance of the FSR sensor is large, which results in the reduction of U_i according to Eq. (1). If $U_i < U_L$, the output U_o will be driven to logic low ($U_{OL} = 0$), and we can calculate the resistance value of R_f by combining Eq. (1), (2) and (4), furthermore, force threshold F_{f1} can be obtained using Fig. 8 (b).

When the foot of the quadruped robot is in contact with the ground and the limb is in stance phase, the resistance of the FSR sensor decreases with the increasing of the GRF, resulting in the increasing of U_i (Eq. (1)). If $U_i > U_H$, the output U_o will be driven to logic high ($U_{OH} = 3.5$ V). Using Eq. (1), (2) and (3), the resistance value of R_f can be calculated, and we can get the relative force threshold F_{f2} using Fig. 8 (b).

Therefore, we can set different force threshold by adjusting the resistance values of R_2 , R_3 , R_4 , and R_5 .

IV. TEST

The flow chart of the control process is shown in Fig. 10:

- (1) As soon as the ARM and DSP processors are powered on or reset, the program based on Linux kernel will be loaded to ARM and the configured program will be loaded to the DSP chip. Then the ARM module waits until it receives the START command from console via Ethernet.
- (2) The ARM processor plans about the quadruped gait, the speed of locomotion according to the obtained sensor information, and sends the planning command to the four DSP processors via SPI.
- (3) The four DSP processors are arranged to control the motion of QR-I's four limbs, respectively. Therefore, based on the ARM command, the four DSP processors separately calculate to generate the desired motor position value, and send the motion control command to the motor drivers via CAN. A loop has been accomplished up to now, and the program goes back to (2), and continues.

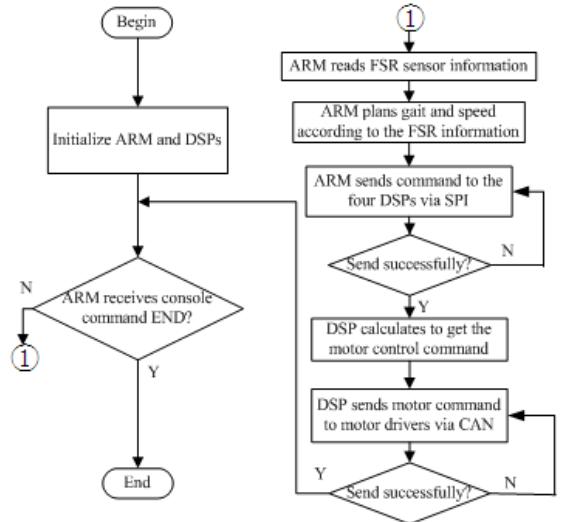


Fig. 10 The flow chart of the control process.

To avoid destroying the DC motor, we test the movement of QR-I's one limb using the designed embedded controller, as depicted in Fig. 11. The limb is in stance phase and the foot is in contact with the ground in Fig. 11(a). Then in Fig. 11(b)-(h), the limb is in swing phase and the foot has no contact with the ground. The limb lifts up in Fig. 11 (b)-(e). With the virtual limb length (the vertical distance between the hip joint and the foot of the same limb) decreases, the limb arrives at the highest point in Fig. 11(e). Afterwards, the virtual limb length increases in Fig. 11 (e)-(h). Finally, the foot touches down the ground in Fig. 11 (i).

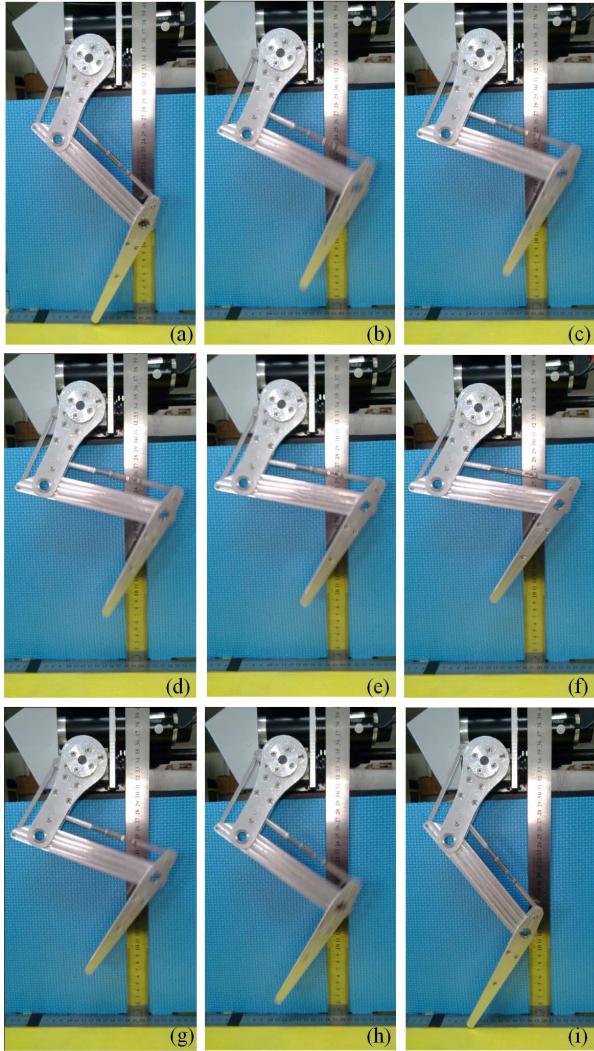


Fig. 11 Snapshots of a video of the the quadruped robot QR-I's one limb moving in a cycle time.

V. CONCLUSION AND FUTURE WORK

In this paper, we design an embedded controller which consists of an ARM board and a DSP board. The ARM board is composed of an ARM processor, interfaces, and the sensor signal modulation circuit. Based on the sensor information, the ARM board plans for switching among gaits, speeding up, and slowing down, and sends the command to the DSP board via SPI. The DSP board contains four DSP processors, SPI, CAN, and JTAG interfaces, and takes charge of generating and transferring the motion control command to DC motor drivers via CAN bus. The embedded controller enables us to achieve independence computation using the ARM and DSP processors, which is adequate for a real implementation of the controller. In addition, we do basic test on the quadruped robot QR-I. The controller is utilized to command the limbs of the quadruped robot moving in swing phase and stance phase. The test result verifies the effectiveness of the controller to realize real-time control.

To avoid irreparable damage, we just do test on one limb of the quadruped robot. Future work will be aiming at developing the quadruped robot QR-I and conducting locomotion experiment on it.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61375101 and 61421004.

REFERENCES

- [1] Marc Raibert, Kevin Blankenspoor, Gabriel Nelson, Rob Playter, the BigDog Team, "BigDog, the rough-terrain quadruped robot," *Proceedings of 17th International Federation of Automatic Control World Congress*, pp. 10822-10825, Seoul, Korea, 2008.
- [2] Murphy M P, Saunders A, Moreira C, et al. "The LittleDog robot," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 145-149, 2010.
- [3] Rebula J R, Neuhaus P D, Bonnlander B V, et al. "A controller for the littledog quadruped walking on rough terrain," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1467-1473, Roma, Italy, 2007.
- [4] Shkolnik A, Levashov M, Manchester I R, et al. "Bounding on rough terrain with the LittleDog robot," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 192 -215, 2011.
- [5] Hyun D J, Seok S, Lee J, et al. "High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah." *The International Journal of Robotics Research*, vol. 33, no. 11, pp. 1417-1445, 2014.
- [6] Semini C. "HyQ-Design and development of a hydraulically actuated quadruped robot," PhD Thesis, University of Genoa, Italy, 2010.
- [7] Semini C, Tsagarakis N G, Guglielmino E, et al. "Design of HyQ—a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 2011.
- [8] Havoutis I, Semini C, Buchli J, et al. "Quadrupedal trotting with active compliance," *Proceedings of the IEEE International Conference on Mechatronics (ICM)*, Karlsruhe, Germany, pp. 610-616, 2013.
- [9] <http://robotik.dfk1-bremen.de/en/research/projects/istruct.html>.
- [10] Kuehn D, Grimminger F, Beinersdorf F, et al. "Additional DOFs and sensors for bio-inspired locomotion: Towards active spine, ankle joints, and feet for a quadruped robot," *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Phuket, Thailand, pp. 2780-2786, 2011.
- [11] Kimura H, Fukuoka Y, and Katabuti H. "Mechanical design of a quadruped "Tekken3&4" and navigation system using laser range sensor," *International Symposium on Robotics*, vol. 36, no. 10, 2005.
- [12] Seok S, Wang A, Chuah M Y, et al. "Design principles for highly efficient quadrupeds and implementation on the MIT Cheetah robot," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3307-3312, Karlsruhe, Germany, 2013.
- [13] Kuehn D, Beinersdorf F, Bernhard F, et al. "Active spine and feet with increased sensing capabilities for walking robots," *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS-12)*, pp. 4-6, 2012.
- [14] Bräunl T. "Embedded robotics: mobile robot design and applications with embedded systems," Springer Science & Business Media, 2008.
- [15] S3C6410X RISC Microprocessor User's Manual. www.samsungsemi.com.
- [16] TMS320F2812 Digital Signal Processor Data Manual. www.ti.com.
- [17] TMS320C1x/C2x/C2xx/C5x Assembly Language Tools [EB]. <http://www.ti.com>.
- [18] SPI Block Guide V03.06. <http://www.motorola.com.cn/>.
- [19] Tindell K, Burns A, and Wellings A J. "Calculating controller area network (CAN) message response times," *Control Engineering Practice*, vol. 3, no. 8, pp. 1163-1169, 1995.
- [20] Davis R I, Burns A, Bril R J, et al. "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239-272, 2007.