An Embedded Control Platform of a Continuous Passive Motion Machine for Injured Fingers

Zhang Fuxiang
Hebei University of Science and Technology
P. R. China

1. Introduction

Human-hand is an organ easy to be injured. Hand-trauma has a high proportion in traumatic cases. About 40 percent of the cases in the emergency treatment of surgery and orthopaedics are hand trauma. Because of the subtle anatomic structure of hands and little muscles all over them, the rehabilitation of the injured hands is a tough task. Normally, the rehabilitation session is long and the recovery of hand function is not quite efficient.

There are mainly three categories of methods for the treatment of hand motion dysfunction during the rehabilitation session. The first is physiotherapy including damp-heat therapeutics, Hertzian waves, ultrasound, He-Ne laser irradiation and faradization. The second is active exercise or passive exercise through using the elastic brace. The third is restoring the injured nerves, relieving the pressures, transplanting and transferring the wholesome muscles and tendons.

Rehabilitation robots are a perfect combination of rehabilitation medicine and robotics. They are not an assistant but an effective means in trauma rehabilitation. In recent years, with the development of Continuous Passive Motion (CPM) theory, CPM machines based on the theory have been used in clinical practice. Former CPM machines mainly aimed at the function training of big joints such as wrists, elbow joints and hocks. Now, CPM machines for the rehabilitation of little joints such as knuckles are available, but they can neither be controlled accurately nor do the function treating of dexterous motions such as grasping and holding. The curative effect needs to be improved. Also, the rehabilitation therapy of the CPM machines rests on the level of empiricism. There are no exact and scientific proofs to prove their curative effect. For the above-mentioned reasons, the purpose of this research is to design a CPM machine for the training of finger function. It can control the motion range, moment and angular speed of the knuckles. It works not only as a device for hand function rehabilitation, but also a means of quantitative detection and evaluation of hand function rehabilitation.

2. Overall System Description

2.1 Scheme of rehabilitation

The main task of the rehabilitation robot in medical practice is to recover the function of the motor system of the injured limbs and trunk. Motor system issues break into two distinct categories: one is related to biomechanical/biophysical applications and the

other to motor learning. Motion-speed and motion-range of a limb is limited by injury, burns, or postoperative conditions in that skin, ligaments, and muscles are inelastic from scar tissue. The biomechanical/biophysical application is to break down scar tissue by using a robotic system to enable greater motion range. The second issue is the learning or possibly the relearning of motor skills. It is complex because it involves a variety of competing motor control theories, training techniques, and human system interaction questions.

Fig.1 shows the schematic diagram of the rehabilitation of the injured fingers by the CPM machine. By combining the feedback information of the finger's motion and the force sent from the sensing system, the functional model information base of human hands and the rehabilitation task, the control information can be obtained. After analysis and calculation, the control information is converted to control signals by the controller and sent to CPM machine to realize the continuous passive motion of the injured fingers.
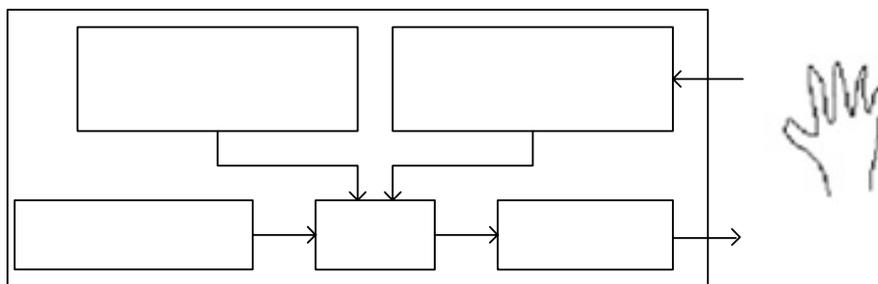

Fig. 1. Schematic diagram of the rehabilitation of the injured fingers.

2.2 Mechanical system

A human hand has redundant DOF. To treat all the joints of a hand is neither practical nor necessary. So the CPM machine is designed for the rehabilitation of knuckles. In order to simplify the structure and achieve modularized design easily, the model is restricted as follows:

1) DOF of the human hand: The finger has 4 DOF, i.e. metacarpo-phalangeal joint (MP) has 2 DOF.
2) Motion range: Although the motion range of the knuckles is different for individuals, it generally has a universal range.
3) Rehabilitation mode: The rehabilitation of phalanges' bending and fingers' abduction/adduction is done respectively.

The CPM machine has three fingers. Each finger is composed of two modules: a biomimetic finger module and a biomimetic muscle module. In order to accurately control the continuous passive motion and make the CPM machine comfortable and convenient to wear, a typical mechanism of the exoskeleton data glove is taken for reference. By using planar four bar linkages and dimensional six bar linkages, the biomimetic finger module can do the driving of one DOF and two DOF respectively.

Because the CPM machine is designed to realize the continuous passive motion for injured fingers, its dimension should be similar to that of the human's fingers. At the present level, it is very difficult to integrate all the drive units into the CPM mechanism. Thus, the CPM machine we designed introduces the technique of biomimetic muscle.

There are two modules in the CPM mechanism for index fingers: biomimetic finger module and biomimetic muscle module. The two modules are linked by biomimetic muscle.

Biomimetic muscles are power sources of the CPM machine (Fig. 2). The CPM mechanism can be driven from long-distance with the biomimetic muscles. So the volume of the CPM mechanism can be reduced greatly. The biomimetic muscle consists of two pulleys, four spring bushings, two springs, a cord and a linear motor. The transmission distance between the linear motor and the CPM mechanism is adjustable by regulating the length of springs and cords.
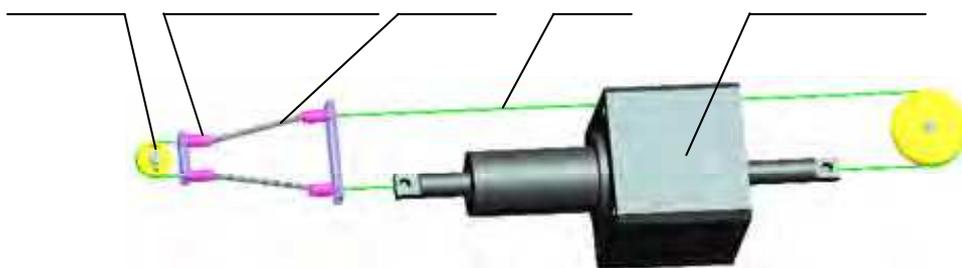

Fig. 2. Biomimetic muscle.

Biomimetic finger module is the execution unit of the CPM machine. Fig. 3 shows the structure of the biomimetic fingers module.
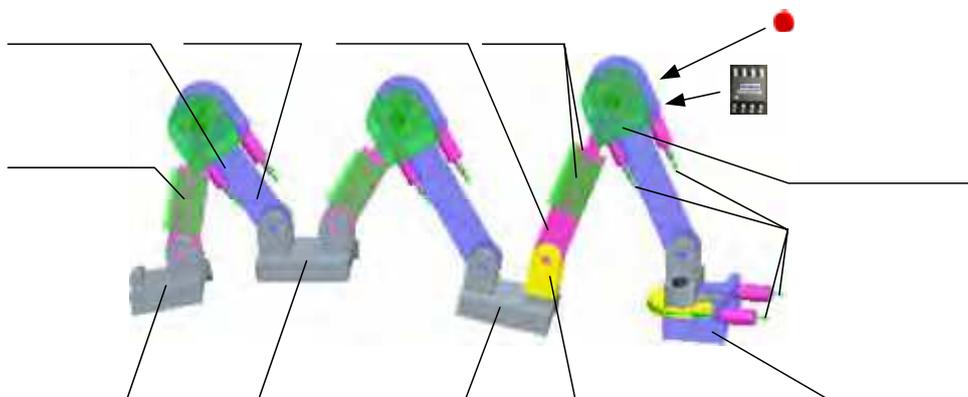

Fig. 3. Biomimetic finger module.

This module by which continuous passive motion of the injured fingers is realized is the main part of the CPM machine. Besides the CPM mechanism, biomimetic finger module also integrates all the joint torque sensors, joint position sensors, sensor signal processing circuit boards and a part of the biomimetic muscles.

Biomimetic muscle module is the drive unit of the CPM machine (Fig. 4). Each biomimetic muscle module has four biomimetic muscles and it can drive four joints of the biomimetic finger module. This module integrates most components of the biomimetic muscles, oriented mechanism, supporting mechanism and tensioning mechanism.
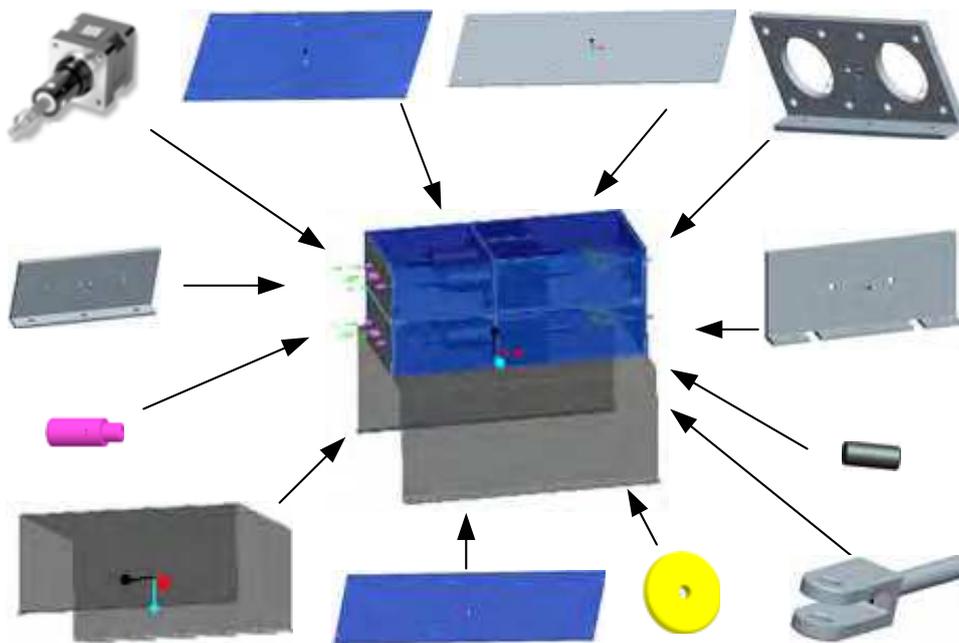
Fig. 4. Biomimetic muscle module.

2.3 Sensing system

Micromation of sensor design requires: 1) The sensors should be a hypostatic union with the CPM machine; 2) Signal processing circuit board should be integrated in the sensor to reduce the disturbance and error generated during signals transmission.

The CPM machine should possess perceptive function to be able to realize rehabilitation training and quantitative evaluation. Integrating joint torque and joint position sensors on the CPM machine can provide essential information to realize the control of continuous passive motion of injured fingers, grasp training etc. The evaluation of clinical rehabilitation will base on the information of the sensors. According to the theory of Evidence-Based Medicine (EBM), better clinical effect can be achieved through the improvement on the control modes and control parameters of the CPM machine.

Strain measurement is a mature and widely used force/torque sensing mode. The basic principle of strain measurement is: At the effect of force or torque, elastic deformation occurs in the elastic body of the sensor. And the resistance value of the strain gages pasted

on the elastic body changes accordingly. Then using bridges to measure the changes of the resistance value, the value of force or torque can be measured. The joint torque sensors used in the CPM machine are specially designed based on the principle of strain measurement. The elastic bodies of the joint torque sensors are integrated in the CPM machine. According to the characteristic of movement transmission and stress in the bars of the mechanism, joints torque sensors are integrated in the motion input bars. The bars integrated with joint sensors are fixed with pulleys, and can be driven by pulley/cord transmission devices. According to the structural features of human fingers, joint torque sensors of distal interphalangeal joints (DIP) and proximal interphalangeal joint (PIP) have 1 degree of freedom (DOF) and those of metacarpo-phalangeal joints (MP) have 2 DOF. Fig. 5 shows the structure of 1-DOF and 2-DOF joint torque sensors. The elastic bodies are cantilevers. They are made of duralumin and have good linearity. 1-DOF and 2-DOF joint torque sensors share similar structure. Their difference lies in that 1-DOF joint torques sensors can only measure the bend torque (see a) in Fig. 5) and 2-DOF joint torques sensors can measure both the bend torque and abduction/adduction torque (see b) in Fig. 5).

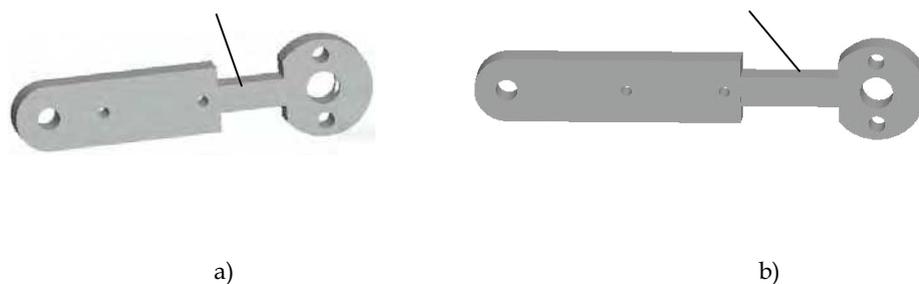a)   b)

Fig. 5. Structure of the 1-DOF and 2-DOF joint torque sensors.

In the joint torque sensors, mini and high-resistance metal strain gages are selected as sense organs to build half-bridge circuits. The circuit boards are fixed on the side surfaces of the bars to meet the demand of micromation and function integration, and to avoid the noise signals generated during long distance signal transmission.

The joint position sensors of the CPM machine can measure the joints' angles while rotating. Getting the right signals of the joint position sensors is the basis of position control of the CPM machine. For the CPM mechanism is very small, dimension restriction is the primary concern when choosing position sensors. So hall position sensors are selected to realize untouched measurement and surface mounted devices (SMD) are used to build signal processing circuit. Thus the micromation of joint position sensors can be realized.

The hall joint position sensor (Fig. 6) fits in the joints of the CPM machine. We use the two-axis Sentron hall sensor 2D-VH-11 as the sensing element of the position sensors. An optimized PCB design exclusively equipped with tiny SMD items plus a circuit with a minimized number of items can make it possible to create a position sensor with excellent performance with respect to its size. The sensor measures only 3.5mm in thickness and 9mm in diameter (Fig. 6). In it fixes a complete analogue conditioning circuit. 12-bit angular solution can be achieved with a linearity error of less than 1% by using a supply voltage of 3.3V.

The working principle of the angular measuring system is described in Fig. 7. A permanent magnet is mounted on the axis of the rotating shaft of every joint, which generates the magnetic field required for the measurement. After offsetting compensation and pre-amplification, the two signals are processed to yield absolute angular position a or angular velocity ω. Extract angular information from the two output signals to obtain the position, angle a can be obtained according to the following function:

$$\alpha = \arctan\left(\frac{V_y}{V_x}\right) \tag{1}$$

By measurement, an angular precision of less than 1° can be obtained in such a design when temperature ranges between -10°C and 60°C and better than 0.3° at constant temperature.
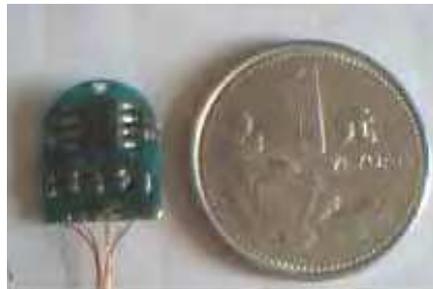

Fig. 6. The joint position sensor.
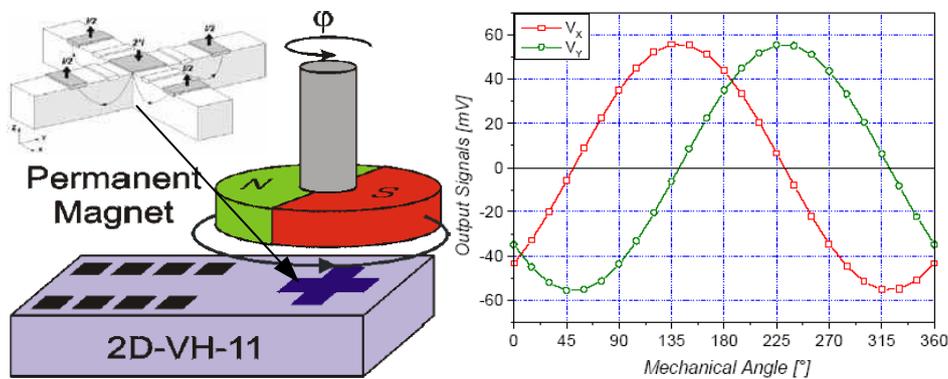
Fig. 7. Operating principle of the hall sensor.

According to the working principle, measuring system needs an alnico to create a magnetic field. The magnetic field created by the alnico should be parallel to the sensor surface. The alnico and the hall sensor 2D-VH-11 are assembled as shown in Fig. 8. When fixing the circuit board on the joint, the rotation shaft should be concentric with the crossing of x axis and y axis on the sensor. The surface of the alnico is parallel to the sensor and perpendicular to the shaft. The alnicos are fixed on the shafts of active joints of the CPM mechanism. On top of the alnicos fix the circuit boards with 2D-VH-11.

Fig. 8. Fixing schematic of the joint position sensor.

3. Overall Design of the embedded System

3.1 Requirements analysis of the embedded system

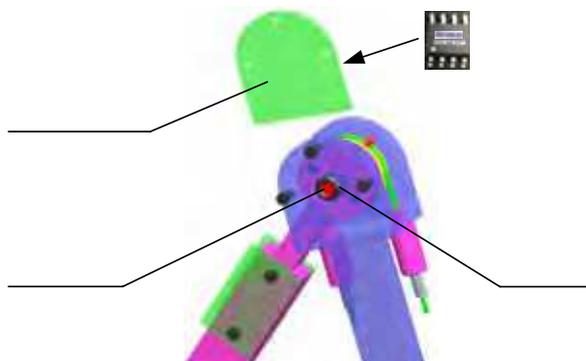According to the clinical experience, thumbs, index fingers and middle fingers rehabilitation training is all-important for the function recovery of human hands, for the main function depends on the three fingers mentioned above. The CPM machine is composed of a thumb, an index finger and a middle finger. The index finger, the middle finger and the thumb have 4, 3 and 1 DOF respectively. Each DOF of the finger is driven by one motor. In order to realize the control task and meet the demand of clinical data acquisition, every joint of the CPM machine is integrated with joint torque and position sensors. Table 1 shows the distributing of the sensors.

Finger/joint	Joint torque sensor number/type/coding	Joint position sensor number/coding
index finger/DIP	1/1DOF/force0	1/angle0y, angle0x
index finger/PIP	1/1DOF/force1	1/angle1y, angle1x
index finger/MP	1/2DOF/force2, force3	1/angle2y, angle2x
middle finger/DIP	1/1DOF/force4	1/angle3y, angle3x
middle finger/PIP	1/1DOF/force5	1/angle4y, angle4x
middle finger/MP	1/1DOF/force6	1/angle5y, angle5x
thumb/IP	1/1DOF/force7	1/angle6y, angle6x

IP- interphalangeal joint
Table 1. Distributing of the sensors.

For the purpose of simplicity and convenience of operation, function requirements are listed as follows:
   1)   The whole CPM machine has three fingers which have a total of 22 analog signals according to Table 1 (a joint torque sensor has one analog signal and a joint position sensor has two analog signals).

2) Low power dissipation.
3) Friendly human-machine interface.
4) Data storage and transfer memory.
5) Expansibility.
6) Embedded hand-held device.

3.2 Specification of the embedded system

According to the above-mentioned function requirements, the technical requirements of the control system are listed as follows:

For the hardware platform

1) Microprocessor and clock rate: SAMSUNG S3C2410, 203MHz.
2) Bus: 32-bit home address bus, 100M bytes bus rate.
3) Memory: 64M bytes 32-bit SDRAM, 16M bytes 16-bit Nor Flash.
4) Display and control mode: 3.5 inch 320×240 TFT LCD, Touch panel.
5) Data acquisition module: A/D convertion of 22-ch analog signals.
6) Motor control module: independent control of 8 motors.
7) Peripheral equipment interfaces: RS-232, USB HOST, Ethernet, JTAG, SD card, etc.

For the software design

1) Operating system: Linux.
2) Driver: LCD & TOUCH Panel, AD Converter and FPGA based on SPI bus.
3) Man-machine interaction: application interface based on graphical user interfaces (GUI)
4) Application program interface (API): sensor signals data acquisition and motors control.

3.3 System integration

The CPM machine is a complex of the CPM mechanism and the embedded system. Fig. 9 shows the system integration of the CPM machine.
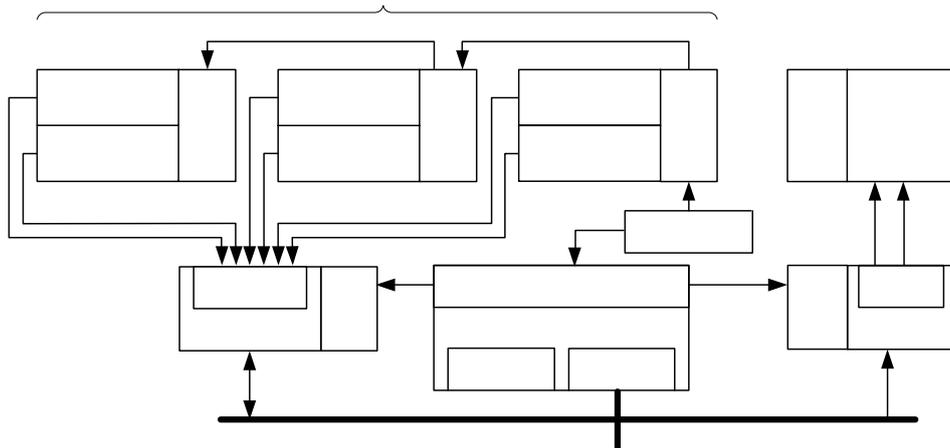

Fig. 9. System integration of the CPM machine.

The sensors and motors of the CPM machine are integrated in the biomimetic finger module and biomimetic muscle module respectively. Restricted by the dimension of the CPM mechanism and the level of micromation of electronic products, the whole system in distributed control structure is difficult to realize. In this system, signal processing of the sensors and drive control of motors are done at the same time. The core of the system is microprocessor S3C2410 whose kernel is ARM 920T. Sense system and drive control system are linked to the SPI bus of S3C2410. All the signals of the sensors are processed by the A/D converters in the data acquisition module. The motors are controlled by a field programmable gate array (FPGA) and several motor controllers.

4. Hardware design of the embedded system

4.1 Methodology of the control platform design
The concept of modularity has been widely used in the design of robot mechanisms to achieve flexibility, economy, ease of maintenance, and rapid deployment. Re-configurable robot is a typical application of modularity design. A fully modular re-configurable robot consisting of a set of standardized modules can be configured to different structures and DOF for different task requirements. The modular approach has been practised in different prototypes built at several research institutes.

Although the method of modularization is widely used in the field of robot design, it rarely touches the field of control platform design. The modularity in above-mentioned robot design is structure modularization. Here we bring forward a concept of function modularization. The control platform can be divided into several modules according to the functions they possess in the system. The core of function modularization is the partition of the control platform. Its main task is to design the modular sub-systems. Any control platform can be divided into several sub-systems through the application of function modularization. The same method helps us achieve feasibility, economical efficiency etc.

4.2 Functional partition of the embedded hardware platform
According to the above-mentioned analysis, we design the schematic diagram of the control platform. Functional modularization is an important characteristic of the control platform as shown in Fig.10.

The control platform is composed of four function modules: an embedded system platform module, a power supply and management module, a motor control module and a data acquisition module. Modularization enables the designers to simplify the design of the control platform and make it applicable for other purposes. The key feature of the control platform is that the motor control module and data acquisition module are linked to a network based on the SPI of the S3C2410. Other function modules can be added to such network to extend the function of the control platform easily.

4.3 Power supply and management module
Power system is a key part of the control platform. The performance of the power system determines the performance of the whole system. The power parameters of the important chips used in the control platform are listed in Table 2. The topology structure of system power (Fig. 11) is designed according to the power parameters.
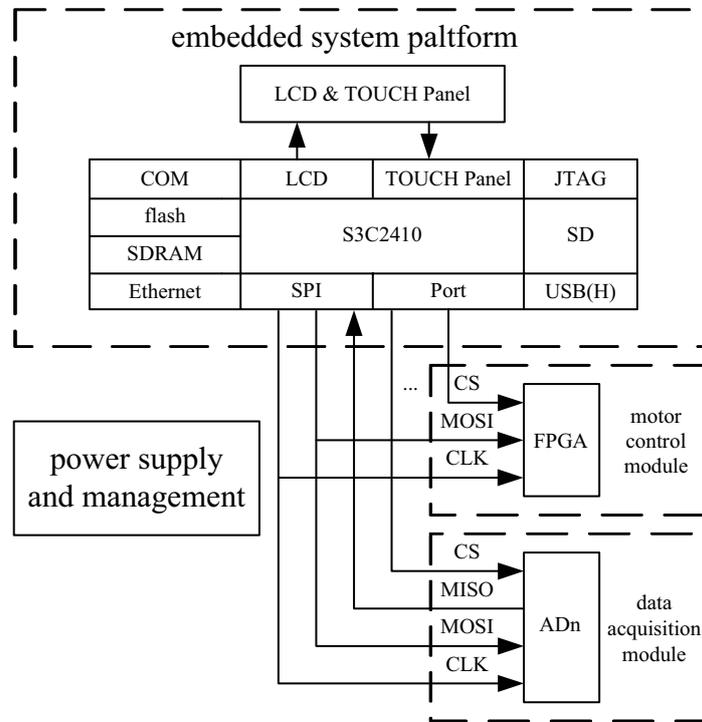
Fig.10. Functional partition of the embedded hardware platform.

Symbol	Count	Operating voltage	Operating current	Power waste
S3C2410	1	1.8V(CORE)	200mA	297mW
		3.3V(I/O)		
SDRAM	2	3.3V	110mA	726mW
FLASH	1	3.3V	80mA	264mW
TFT LCD	1	+5V(VSHA)	6mA	30mW
		+3.3V(VSHD)	3.5mA	11.55mW
		+15V(VDD)	0.1mA	1.5mW
		-10V(VEE)	-0.1mA	1mW
		21.6V(Back light)	20mA	432mW
DM9000	1	3.3V	100mA	330mW
AD7888	3	3.3V	0.64mA	6.3mW
AD623	22	3.3V	575μA	41.7mW
EPF10K10A	1	3.3V	12mA	39.6mW
SP3232E	1	3.3V	100mA	330mW
BLCPS	8	6V	100mA	4.8W
USB	1	5V	0.4mA	2mW

Table 2. The power parameters of the important chips.

Fig.11. Topology structure of the power supply system.

AC adapter (input: 220V, output: 7.5V, 3A) is used in the power supply system. The whole system selects voltage references with low voltage difference and low noise. Battery feed is also optional in the design. When using battery as power input, in order to improve the efficiency of the system, a LT1085 can be used to boost the voltage to 5V for the motor controllers and above 6.5V for the rest part of the system.
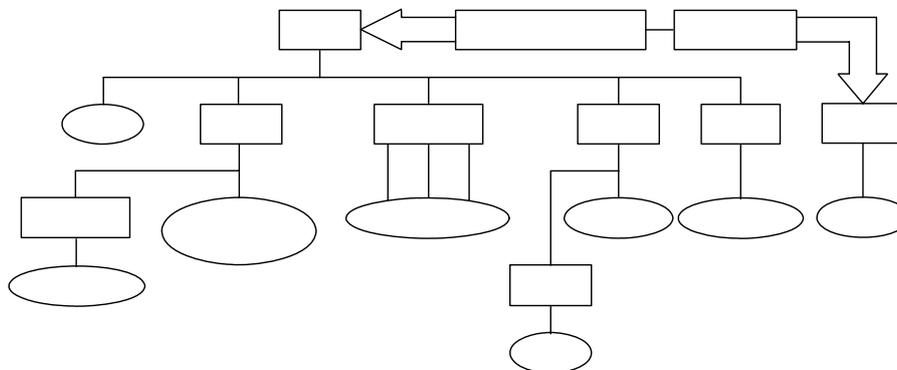
4.4 Embedded system platform

The embedded system with specially designed S3C2410 as its core, which is applicable to PDAs, palmtop computers and GPS devices, is the core part of the control platform. Its design requirement can be met by adding the necessary peripheral circuit and the corresponding peripheral equipments. The software platform of the system is an embedded Linux operating system.

The embedded system platform is composed of three function modules: a core board for minimum system of S3C2410 which consists of CPU, FLASH and JTAG, an expansion board of the minimum system of S3C2410 which integrates the interfaces of RS-232, USB HOST, Ethernet, JTAG and SD card, a board for LCD & TOUCH Panel.

4.5 Data acquisition module

Data acquisition module is one of the function modules linked to the SPI network. Data acquisition is completed by some AD7888s, each of which can process 8-ch serial data. The number of processing signals decides the number of AD7888 in demand. All the AD7888s are linked to SPI bus. We can process more signals by simply adding more AD7888s to the SPI network. The I/O ports of the S3C2410 complete the selection of the channels to be sampled. If the number of chips linked to the SPI network is too large, we should add an encoder to the circuit to reduce the occupation of the I/O ports.

4.6 Motor control module

The motor control module is composed of several motor-controllers and FPGA. Each motor controller controls a mini linear motor. The motors used in the module are 35H4N-2.33 made by the Haydon Linear Motors Company (Fig. 12, captive shaft, main parameters: continuous thrust 230N, travel 25.4mm). The motor controllers used in the module are DCM4010 specially designed for mini linear motors made by Haydon Linear Motors Company. DCM4010 is a subdivision driver with high performance which selects the mode of ambipolar DC chopper and fits for the drive control of two phase composite step motors (Fig. 13).

Fig. 12. Linear stepping motor 35H4N-2.33.

Fig. 13. Driver DCM4010.

Interface definition of DCM4010 shows in Table 3.

Signals	Functions
PUL	pulse signal: rising edge is available
DIR	direction signal: TTL
OPTO	light coupling power supply
ENA	enable signal: low is forbiden
GND	ground
+V	power supply (+24V)
A+	phase A+
A-	phase A-
B+	phase B+
B-	phase B-

Table 3. Interface definition of DCM4010.

The whole motor control system consists of step motors, drivers, dc supply and controllers. In this module, the 8 step motors are controlled by a FPGA (Fig. 14).
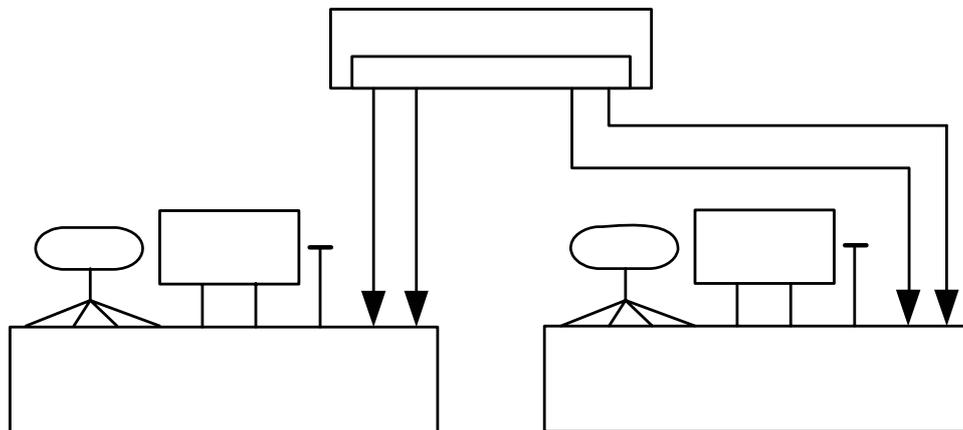

Fig. 14. Motor control system.

The pins of the FPGA should be defined after its hardware is build up. Definition of the pins of the FPGA except those that are correlative to the system is shown in Table 4. *f0-f7* are the controlled output frequencies, *dir0-dir7* are the direction of the motors. We can achieve the speed and direction control of the motors through sending commands to the ports. *SCLK* is the input clock, *MOSI* is the input data, and *F2M* is the pulse input by a crystal oscillator with 2MHz output. *CS* is the chip selection of the FPGA.

The motor control module is linked to the SPI bus by the FPGA. Here, the impulse generator of the FPGA adopts top-down design. Module *SPI_Motor* is at the top, which can be further divided into *SPI_Core* and *Counter*. The system division of the circuit design is shown in the Fig.15. We can add more motor controllers to control more mini linear motors by changing the program of impulse generator in the FPGA.

Serial number of the pins	Function definition	Serial number of the pins	Function definition
Pin61	dir0	Pin58	f0
Pin57	dir1	Pin56	f1
Pin71	dir2	Pin70	f2
Pin69	dir3	Pin68	f3
Pin45	dir4	Pin44	f4
Pin43	dir5	Pin42	f5
Pin65	dir6	Pin64	f6
Pin63	dir7	Pin62	f7
Pin21	MOSI	Pin15	CS
Pin19	SCLK	Pin	F2M

Table 4. Definition of the pins of the FPGA.

Fig. 15. Modules of the motor control system.

Flow chart of the FPGA shows in Fig. 16. After the system is electrified, the FPGA is initialized partly. That is to say, it is the registers used in *SPI_Core* module not in the *Counter* module that are initialized. The un-initialized registers will be initialized in the drivers of motors. After the FPGA is selected, *SPI_Core* module begins to receive data from SPI bus. One byte data is sent at one time. After the one-byte data is received, *SPI_Core* module will do a serial-parallel conversion. Because a control signal of motors consists of two bytes, it is necessary to add a command during the serial-parallel conversion to determine whether the conversion is completed. If the conversion is completed, the *SPI_Core* module will inform *Counter* module to start receiving the data. *Counter* module begins to work according to the initial value of the registers after the system is electrified. After receiving the converted commands, *Counter* module will send the data to the relevant registers according to the address information in the commands. Every bit of the *Dir* register has an output port which is linked to the relevant direction terminal of the motor drive circuit board. The changes of *Dir* register lead to the changes of direction of the relevant motors.

Fig. 16. Flow chart of the FPGA.

The *T_Enable* register and frequency division counters work cooperatively. Only when the *T_Enable* register enables some motors to work, can the frequency division counters of the relevant motors begin to work. The frequency division counters are initialized before being enabled by the *T_Enable* register. After being enabled by the *T_Enable* register, relevant frequency division counters begin to divide the frequency. During frequency division, the value of counters should be checked. If they reach the set value, the output levels will be reversed. At the same time, the counters are adjusted to zero. In this way, the speed control of the motors is realized.

The function of the *Counter* module is to control the speed of the motors. The *Counter* module is made up of 8 timers, an encoder and a controlled frequency divider. The output frequencies of the *Counter* module is

$$F_{out} = \frac{2000000}{(2n_1 + 2) \times (2n_2 + 2)} \tag{2}$$

In the above function, n_1 and n_2 are the coefficient of the frequency divider and the value of the timer register respectively. The range of n_1 is between 0 and 255, and the range of n_2 is between 332-4095. The output frequencies of the *Counter* module are shown in Fig. 17.

The *SPI_Core* module receives the data from MOSI data bus, which comes from the SPI interface in S3C2410, and completes the serial-parallel conversion task, then transfers the converted data to the *Counter* module. According to the design purpose, the *SPI_Core* module mainly consists of a serial-parallel conversion task, a state register and an address register. In this design, the front 4 bits of the first byte is the address of the corresponding register; the back 4 bits of the first byte and the second byte are the data sent to the corresponding register. The SPI interface can transfer one byte at one time, whereas a complete control requires two bytes. *Data_F* is used to denote the number of received bytes. *EOC* is used to adjust the *SPI_Core* and the *Counter* module.

The serial-parallel conversion task is completed by a finite state machine, which can convert the serial data in MOSI bus to parallel data. The state transition graph is shown in Fig.18. As it shows, *Sh8 in_bit7* is the first state. State encoding adopts one hot coding. This encoding method uses n triggers to achieve n state of state machine. Although the number of triggers is large, many assembled circuits can be saved. It results in less complex circuit and much improved reliability. To the FPGA, which has much sequence-ordered, less combined logic information, this is a better encoding method.
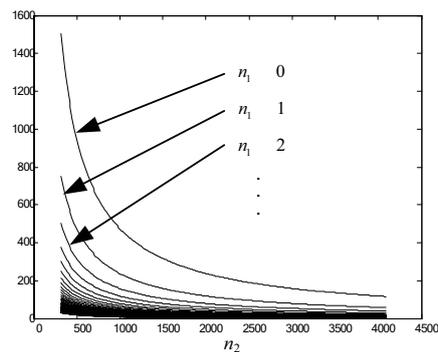

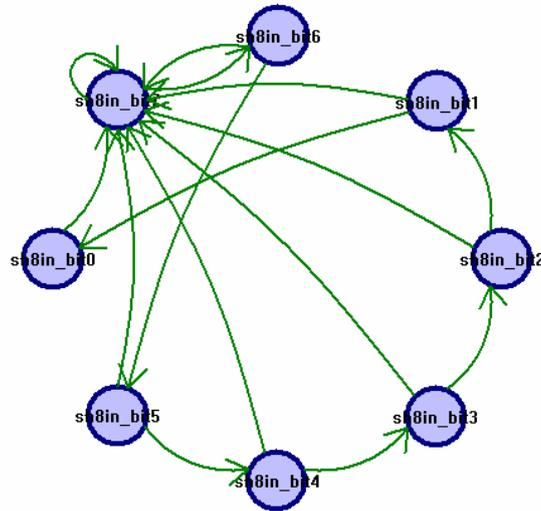Fig. 17. Output frequencies of the Counter module.

Fig. 18. The state transition graph of the serial-opparallel conversion.

The *ADD_Encode* address register does the coding of the transferred object. Detailed coding is shown in Table 5.

Input data (Binary system)	Registers	Function description
0000	T_Enable	Enable bit: 0 startup, 1 stop
0001	Dir	Direction of the motors
0010	T0	Timer0
0011	T1	Timer1
0100	T2	Timer2
0101	T3	Timer3
0110	T4	Timer4
0111	T5	Timer5
1000	T6	Timer6
1001	T7	Timer7

Table 5. Table of internal registers.

Synthesizing the VerilogHDL code by Synplify Pro7.2, we get the netlist file in format of EDIF. Using the QUARTUS4.1 we can do placing and routing, get the sequential simulation files and downloaded files by analysing the netlist. Its sequential simulation file is downloaded into EPF10K10TC100 and EPC2 by BYTEBLASTER. Fig. 18 shows the oscillogram of the *SPI_Motor* module. In the simulation program, the value of *T_Enable* is 1111_0000. It denotes that motor 0, 1, 2 and 3 are enabled to rotate. f_0, f_1, f_2 and f_3 are the speed output of the motors. The directions of the motors are set by Dir=10101010. F3K is the clock after frequency division. T0=T1=0000_0000_0001 and T2=T3=0000_0000_0101 denotes that the speed of motor 0 is equal to that of motor 1 and the speed of motor 2 is equal to that of motor 3 respectively.
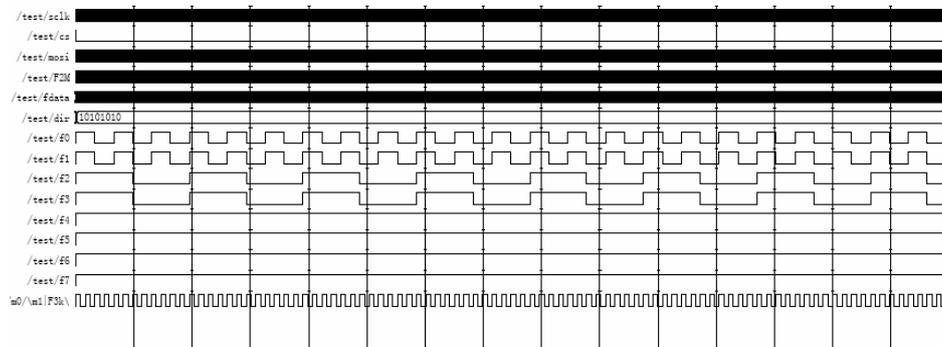
Fig. 19. Oscillogram of the SPI_Motor module.

5. Software design of the embedded system

5.1 Software structure of embedded systems

Software structure of embedded system of the CPM machine is shown in Fig. 20. Considering the need of project research and product development in the future, free embedded Linux is selected as the operating system (OS) for the system. Main task of the embedded system development is to provide the application software run on the OS. The application software consists of drivers, GUI, control software and API. Control programs consist of a main program and several subprograms: The main control program runs monitors and controls the whole system; the subprograms process the sensors' data and implement the control strategy and each control method.
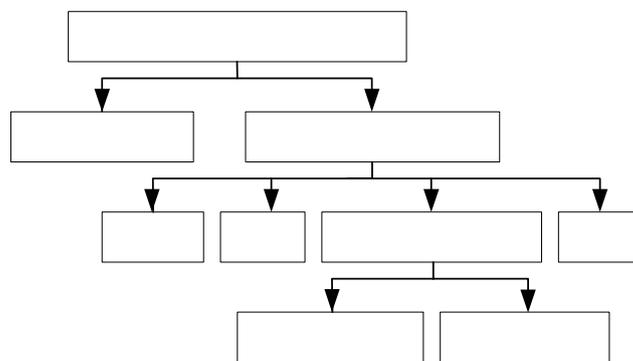

Fig. 20. Software structure of the embedded system

5.2 Hierarchical parallel competitive control architecture

The CPM machine system selects the hierarchical parallel-competitive control structure (Fig. 21). In this structure, low-grade modes are restricted by high-grade modes. If a high-grade mode is not enabled, the system will select the enabled low-grade mode. If the high-grade mode is enabled, the low-grade mode will be restricted and the high-grade mode will control the system until a higher grade mode is enabled.
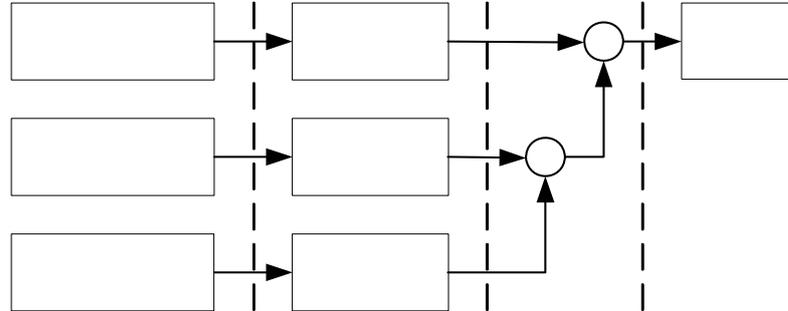
Fig.21. Hierarchical parallel-competitive control structure

Introduce the modes of the system by priority:
Mode 0 (emergency state mode): This mode is used when power supply voltage is out of gauge, motion of the CPM machine is blocked or during urgent stop etc. In this case, the system will disable all low-grade modes and show the graphical and literal alarm information by GUI.
Mode 1 (manual mode): This mode is used for manual control.
Mode 2 (programmed control mode): In this mode, rehabilitation parameters are set through the GUI and the rehabilitation motion is completed automatically.

5.3 Drivers of the embedded system
All peripheral equipment of the embedded system has controllers. In Linux, the codes which manage the hardware controllers are managed by the kernel. The software which manages the hardware controllers is device drivers. The kernel communicates with the peripheral equipments via their device drivers. A device driver as one part of the kernel is a set of data structures and functions. These data structures and functions control one or more devices by defined interfaces. To the user programs, the device drivers map the peripheral equipments into device files. User programs can dispose the device files as common files.
A device driver consists of three parts: initialization, device-independent interfaces and hardware I/O. Hardware I/O is interrelated with the hardware. There are three modes which can realize data switching between the CPU and the hardware: polling, DMA and interruption. Device-independent interfaces work as bridges between device drivers and the file system. Data structure *file_operations* is a normative interface in Linux. Initialization realizes the loading and unloading the device drivers from the kernel. For character devices, the initialization functions are run in */drivers/char/mem.c* in the Linux source codes. For block devices, the initialization functions are run in */drivers/block/ll_rw_block.c* in the Linux source codes. A device driver which can be loaded dynamically has two functions: one is entry point function *init_module( )* transferred by Linux system command *insmod*; the other is exit point function *cleanup_module( )* transferred by Linux system command *rmmod*.
Two SPI ports, provided by the ARM chip S3C2410, enable high-speed serial transmission of the data. The initialization process of SPI bus is as follows:
    1)   Set SPI Baud Rate Prescaler Register (SPPREn);
    2)   Set SPI Control Registers (SPCONn) to configure the SPI module;

3) Write data 0xFF to SPI Tx Data Register (SPTDATn) 10 times in order to initialize the devices linked to the SPI bus;

4) Set a GPIO pin, which acts as nSS, to low to activate the devices linked to the SPI bus.

In this system, three AD7888s and a FPGA are linked to the SPI bus. Table 6. shows the distribution list of general I/O ports.

I/O prots	Input/output	Function
GPH4	OUTPUT	CS of AD7888 (AD0)
GPH5	OUTPUT	CS of AD7888 (AD1)
GPH6	OUTPUT	CS of AD7888 (AD2)
GPH7	OUTPUT	CS of FPGA

Table 6. Distribution list of general I/O ports.

The SPI registers used in the drivers are rSPCON0, rSPSTA0, rSPPIN0, rSPPRE0, rSPTDAT0 and rSPRDAT0. The port registers are rGPECON, rGPEUP, rGPGCON, rGPGUP and rGPGDAT. (r denotes register)

SPI bus is driven by the initialization function *Init_SPI( )*. Its pseudo codes are:

    Set rSPPRE0   (set baud rate of the SPI bus of S3C2410)

    Set SPCON0   (select polling mode)

    For i=1 TO 10

       Set rSPTDAT0,0xff   (initialize the devices linked to the SPI bus)

    Set rGPECON   (set pins 4-7 of E port for SPI function)

    Set rGPEUP   (forbid pull-up resistor function of E port)

    Set rGPHCON   (set 4-7 pins of H port for output)

    Set rGPHUP   (enable H port for pull-up resistor)

Since polling mode is selected, the state of the SPI should be checked before data transfer. The function is realized by *spi_poll_done( )*. Its pseudo codes are:

    while do

      rSPSTA0&0x01   (if SPI bus is busy)

Data transfer through SPI bus is realized by function *spi_tx_date( )*. Its pseudo codes are:

    spi_poll_done();

    Set rSPTDAT0,data   (transfer data through SPI bus)

    spi_poll_done();

In this system, three AD7888 (AD0, AD1 and AD2) are used. For the three drivers, only main device number, main device name and the port for chip selection are different, the process and methods used in the drivers all goes the same way.

A/D convert of ADn is realized by function *adn_convert( )*. Its pseudo codes are:

    Set rGPHDAT,CS   (select ADn)

    spi_tx_data(ADnTXdata[0])   (select power management mode and channel of ADn)

    Read ADnRXdata[0],rSPRDAT0   (read 4 higher bits of the results of A/D convertion)

    spi_tx_data(0xff)   (send dummy '1' data to start A/D convertion)

    Read ADnRXdata[1],rSPRDAT0   (read 8 lower bits of the results of A/D convertion)

    Set rGPHDAT,unCS   (unselect ADn)

Device file structures of ADn are:

```
static struct file_operations adn_fops={

owner:THIS_MODULE,

read:adn_rd,

write:adn_wr,

open:adn_open,

release:adn_close,

};
```

System transfer methods used in the device file stuctures of ADn are read, write, open and release. Pseudo codes of read method *adn_rd( )* are:

    adn_convert()   (start A/D convertion: for channel sellection)

    adn_convert()   (start A/D convertion: for A/D convertion)

    Set dbuf,kmalloc(2*sizeof(unsigned char),GFP_KERNEL)   (allot data buffer)

    For i=0 TO 1

    Read dbuf[i],ADnRXdata[i]   (read data from SPI bus to data buffer)

    copy_to_user(buf,dbuf,count)   (read data from data buffer to user space)

kfree(dbuf)   (release data buffer)

Pseudo codes of write method *adn_wr( )* are:

    Set dbuf,kmalloc(sizeof(unsigned char),GFP_KERNEL)   (allot data buffer)

    copy_from_user(buf,dbuf,1)   (read data from user space to data

                                                    buffer)

    Set ADnTXdata[0],dbuf[0]   (send data from data buffer to SPI

                                                    bus)

    kfree(dbuf)   (release data buffer)

Open method *adn_open( )* and close method *adn_close( )* are do nothing functions. The methods of loading and unloading function *adn_init( )* and *adn_exit( )* of ADn are the same as those of common character devices.

One FPGA is used in the system to control 8 motors. Sending data to the FPGA is realized by function *motor_data_send( )*. Its pseudo codes are:

    Set rGPHDAT,CS   (select FPGA)

    For i=0 TO 1

      spi_tx_data(TXdata[i])   (send motor control commands to registers

                                         of FPGA)

    Set rGPHDAT,unCS   (unselect FPGA)

For the motors are operating elements of the CPM machine, the driver module of FPGA should initialize the state of the FPGA to avoid the motors' malfunction. The function is realized by function *Init_Motor ( )*. Its pseudo codes are:

    spi_tx_data(MOTOR_Ena)   (select T_Enable register of FPGA)

    spi_tx_data(Ena_Start)   (intialize T_Enable register: enable the

                                          appointed motors to work)

    spi_tx_data(MOTOR_Dir)   (select Dir register of FPGA)

    spi_tx_data(DIR_Start)   (intialize Dir register)

    spi_tx_data(MOTOR_Tn)   (select T0-T7 register of FPGA)

    spi_tx_data(Tn_Start)   (intialize T0-T7 register of FPGA)

    spi_tx_data(MOTOR_Stop)   (select T_Enable register of FPGA)

spi_tx_data(Ena_Stop)   (intialize T_Enable register: disable the

appointed motors to work)

Device file structure of FPGA is:

    static struct file_operations motor_fops={

    owner:THIS_MODULE,

    write:motor_wr,

    open:motor_open,

    release:motor_close,

    };

The system transfer methods used in device file structure of the FPGA are write, open and release. Because no data needs reading from the FPGA, no read method is used in the device dirver of the FPGA. Other methods used in the device dirver of the FPGA are the same as those of ADn.

Close method *motor_exit( )* of the FPGA works the same way as those of common character devices. In the open method *motor_init( )* a function *Init_Motor ( )* is added to realize the initialization of the motors.

5.4 Graphical user interfaces of the embedded system

For the hardware based on 32-bit embedded processors has high-speed and large capacity internal memory, it is preferred to build GUI for human-machine interaction. The GUI of the CPM machine is developed with MiniGUI, which was specially designed for GUI design under Linux by Feynman Company.

The GUI of the CPM machine is in hierarchical structure (Fig. 22). The main interface has 4 main functions: rehabilitation training, system setting, games and help. Rehabilitation training and system setting are main function modules of the GUI. Rehabilitation training has two modes: CPM and H-CPM. In CPM mode, rehabilitation training of bend and wiggle is done respectively. The system setting completes two tasks: one is to set the rehabilitation parameters used in the rehabilitation training; the other is to save the joint torque/position information to the flash disk.
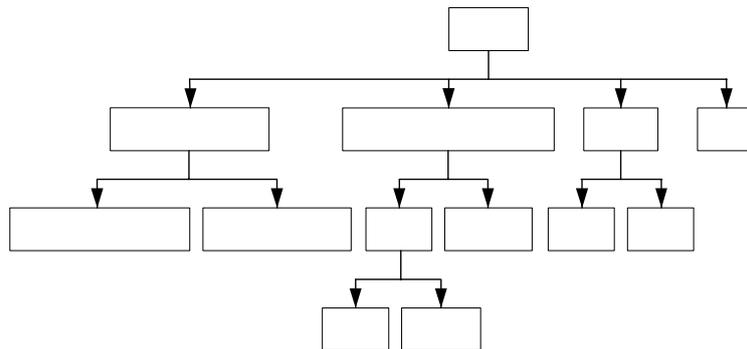
Fig. 22. Structure of the graphical user interfaces.

5.5 Application programming interface

In the CPM machine system, Linux kernel is highly reduced and Linux C programming language is selected. To the special devices such as the FPGA and AD7888s, there are no API to use. In order to develop flexible, applied and efficient programs, it is necessary to develop API for the FPGA and AD7888s. The API functions link the drivers and the control program. Sensor data acquisition and motors control by the FPGA are realized by reading and writing corresponding device files. Packaging the process mentioned above can get the API functions. Sensor data is collected by reading the information of the AD7888's channels. Table 7 shows the allocation of AD channels.

Channels	1	2	3	4	5	6	7	8
AD0	angle0y	angle0x	angle1y	angle1x	angle2y	angle2x	force0	force1
AD1	force2	force3	angle3y	angle3x	angle4y	angle4x	angle5y	angle5x
AD2	force4	force5	force6	angle6y	angle6x	force7	—	—

Table 7. The allocation table of AD channels.

Here the API function of joint position sensor *angle0y* is taken as an example. It is realized by the function *SampleAngle0y( )*. Its pseudo codes are:

```
Set Send[0],0x04                    (select no. 1 channel of the AD0, mode 0)

Set fd,open("/dev/ad0",O_RDWR)      (open device file of AD0)

write(fd,&Send[0],1)                (send command of data acquisition)

read(fd,AD01,2)                     (read the results of data acquisition)

close(fd)                           (close device file of AD0)

Report result                       (return data acquisition results)
```

Direction and speed control are realized by sending data to the FPGA. It is realized by the function of *MotorGoto( )*. Its pseudo codes are:

```
Set fd,open("/dev/motor",O_RDWR)    (open device file of FPGA)

Set Start[0],0                      (set address of T_Enable register)

Set Start[1],Tenable                (set contents of T_Enable register)

write(fd,Start,2)                   (send Tenable command to FPGA)

Set Dir[0],0x10                     (set address of Dir register)

Set Dir[1],Dir                      (set contents of Dir register)

write(fd,Dir,2)                     (send Dir command to FPGA)

Convert n and f to H and L          (convert frequency motor n into 2-byte data)

Set Out[0],H                        (Tn register address + 4 higher bits of the
                                    contents)
```

Set Out[1],L   (Tn register 8 lower bits of the contents)

write(fd,Out,2)   (send motors speed control commands to FPGA)

write(fd,Start,2)   (send Tenable command to FPGA, start the
                                               motors)

close(fd)   (close device file of AD0)

5.6 Control program

Fig. 23 shows the diagram of the control program. The control program integrates all the programs including OS. It can complete initialization of the software platform, process control and system supervision of the CPM machine.

Fig.23. Diagram of control program.

6. Testing

6.1 Aims of the rehabilitation motion tests

The aims of the rehabilitation motion tests are: 1) Validate the maneuverability, comfort and security; 2) Validate the function of data collection of the system; 3) Analyse biomechanics characteristic of healthy human hands during rehabilitation motion.

In order to reach the goals mentioned above, rehabilitation motion tests have been made on three volunteers and rehabilitation data of the tests has been collected for later research. Considering the cost factor, only the CPM mechanisms for the index finger are manufactured for the first prototype of the CPM machine.

6.2 Instruction of volunteers

volunteer A: male, 33 years old, length of knuckles: P3=22mm, P2=30mm, P1=48mm;
volunteer B: male, 24 years old, length of knuckles: P3=23mm, P2=31mm, P1=46mm;
volunteer C: female, 24 years old, length of knuckles: P3=19mm, P2=26mm, P1=45mm.

6.3 Rehabilitation motion test of healthy human hands

Since modularization design is adopted in the mechanism of the CPM machine, the CPM machine can do rehabilitation training through different combinations as: 1) IP of the thumb; 2) DIP of the index finger or the middle finger; 3) PIP of the index finger or the middle finger; 4) MP of the index finger or the middle finger; 5) DIP and PIP of the index finger or the middle finger; 6) PIP and MP of the index finger or the middle finger; 7) DIP and PIP and MP of the index finger or the middle finger.
Fig. 24 shows the CPM machine during rehabilitation training.
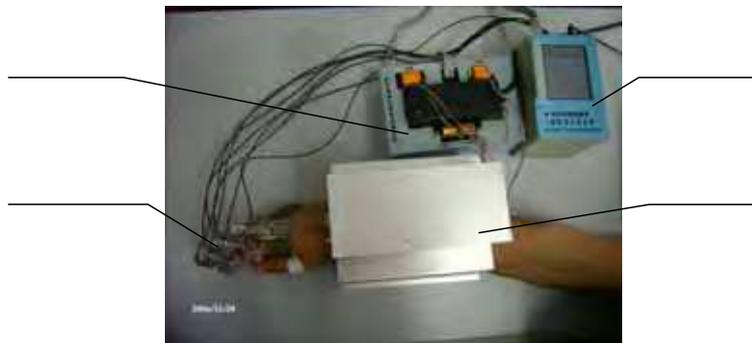
Fig.24. Rehabilitation training using the CPM machine

According to the text condition, rehabilitation motion tests on DIP of volunteer A and B and rehabilitation motion test on PIP of the volunteer C have been made. Test data has been collected during the rehabilitation motions. The basic settings are: sampling period = 100ms; safety torque = 150mNm. Fig. 25 shows one cycle of the rehabilitation motion.

a) Bend

b) Extend
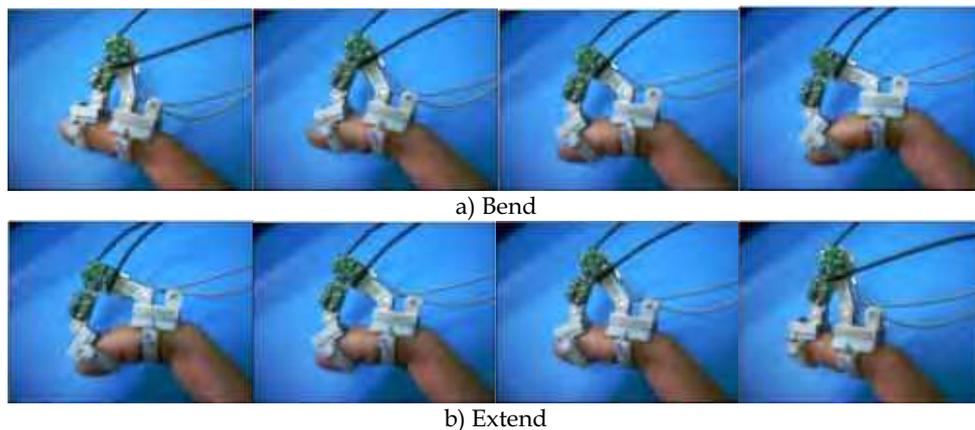
Fig. 25. Rehabilitation of single joint.

Relationship of joint position and joint torque of the volunteer A, B and C during a certain cycle is shown in Fig. 26 a), b) and c) respectively. The curves in Fig. 26 show that: 1) Because the curves of the joint position and joint torque are close and the relevant come-and-go curves are superposed roughly, the curves basically reflect the biomechanics characteristic of the joints; 2) There are exceptional knots in the curves (nearside of a) and b) and starboard of c)) because the flexibility of the cords results in delay when the motors are reversed; 3) The come-and-go curves are not in well superposition because the frictional resistances of the cords in the relevant come-and-go course are different; 4) The maximal values of joint torques are at the position that the joint angles are the smallest because here is the bend limit position of the fingers. To break the restriction of the motion in this direction and reach or close to the physiological limit position is the aim of rehabilitation training. Joint position settings in this direction are main technical parameters of rehabilitation training.

a)   b)   c)
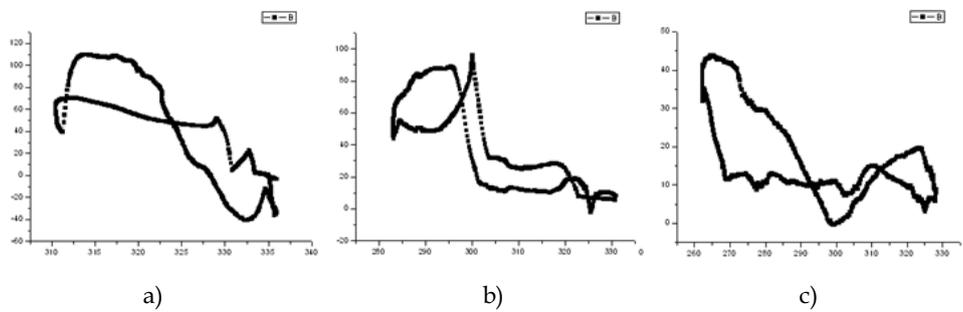
Fig. 26. Relationship curves of joint position and its torque during rehabilitation motions.

In order to evaluate the performance of the whole system, multi-joints (include DIP, PIP and MP) rehabilitation motion test is done on volunteer A (Fig. 27). During the test, the control operation is simple and convenient; the rehabilitation motion is steady and reliable. Volunteer A feels cosy too.
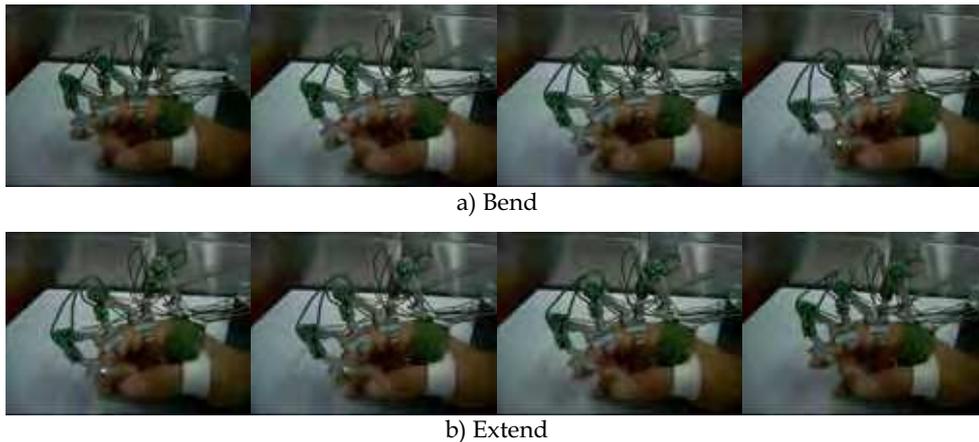

a) Bend

b) Extend

Fig. 27. Rehabilitation of multi-joints.

For the CPM machine, users can adjust the ranges of the rehabilitation motion and set the rehabilitation cycle within a certain range during the training. According to the survey, the rehabilitation cycle can reach 20s when the subdivision number of the DCM4010 is set to 8. We can reduce the rehabilitation cycle to 10s only by setting the subdivision number of the DCM4010 to 4. The CPM machine is applied to the early function rehabilitation of injured fingers. According to the experience of the rehabilitation physician, early rehabilitation training should not go too quickly, that is to say, the lowest rehabilitation cycle is not less than 15s. Considering the motion range of the injured joints is restricted, that the subdivision number of the DCM4010 is set to 8 can meet the demand of most clinical applications.

7. Conclusion and future work

We bring forward the design philosophy of function modularization design for control platforms. With this thinking we design a control platform, which is composed of four function modules: an embedded system platform module, a power supply and management module, a motor control module and a data acquisition module. In the control platform, the embedded system platform provides a basic platform for different applications. The embedded system platform uses the ARM chip S3C2410 as its core, and embedded Linux as the OS. The embedded system platform integrates the most common interfaces of the microprocessors such as RS-232, USB HOST, Ethernet, JTAG and SD card etc. The dada acquisition module and the motor control module are based on the SPI network, which is the key feature of the system. Power supply and management module provide the power for the whole system. The whole control platform is open ended for new functions and applications if new chips are added into the SPI network.

The first prototype has been finished. We have made some system test on several normal volunteers and Soon we will do patient testing in Harbin Medical University. We will make a series of clinic experiments to explore the controlling method applicable to the CPM machine. We plan to test the acceptance of the system and get more data of the rehabilitation of injured fingers at different cases as well. By using the theory of CPM as the guidance for the training of injured fingers and the theory of EBM as the enrichment to their training methods, we can make the CPM machine better meet the requirement of the rehabilitation of injured fingers, and at the same time promote the development of rehabilitation theory.

8. References

O'Driscoll, S. W.; Giori, N. J. (2000). Continuous passive motion (CPM): theory and principles of clinical application. *J Rehabil Res Dev*, Vol. 33, No. 2, (Feb 2000) 179–188, ISSN

ZHANG, F. X.; FU, Y. L.; WANG, S. G. (2005). An overview of rehabilitation robots research. *Hebei Journal of Industrial Science and Technology*, Vol. 22, No. 2, (Mar 2005) 100-105, ISSN

Khalili, K.; Zomlefer, M. (1988). An intelligent robotic system for rehabilitation of joints and estimation of body segment parameters. *IEEE Transaction on Biomedical Engineering*, Vol. 35 (Feb 1988) 138-146, ISSN

Wang, J. S.; Wang, T. M.; Wei, J.; Han, Z. Z.; You, S. (2000). Development of a new telemanipulation-oriented data glove. *Robot*, Vol. 22, No. 3 (Mar 2000) 201-206, ISSN

Wright, A. K.; Stanisic, M. M. (1990). Kinematic mapping between the EXOS Handmaster exoskeleton and the Utah/MIT dexterous hand, *Proceedings of IEEE International Conference on Systems Engineering*, pp. 101-104, ISBN, Pittsburgh, PA, USA, Aug 1990, Pittsburgh

Zhang, Q. X. (1984). *Analysis and synthesis of dimensional linkages*, China Machine Press, ISBN, Beijing, China

Liu, H.; Hirzinger, G. (2003). DLR dexterous robot hand I and II. *Journal of Xi'An Jiaotong University*, Vol. 37, No. 4 (Apr 2003) 331-337, ISSN

Lee, J. J.; Lee, Y. Y. (2003). Dynamic analysis of tendon driven robotic mechanism. *Journel of Robotic System*, Vol. 20, No. 5 (Apr 2003) 229-238, ISSN

Liang, B.; Gao, X. H.; Jin, M. H. et al. (2006). Recent progress of the chinese intellignent space robotic system, *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, ISBN, Bejing, China, (Oct 2006) pp. 994-1001, Beijing

Karbasi, H.; Huissoon, J. P.; Khajepour, A. (2004). Uni-drive modular robots: theory, design, and experiments. *Mechanism and Machine Theory*. Vol. 39 (2004) 183-200, ISSN

Xu, H.; Wang, S. G.; Fu, Y. L.; Li, H. (2005). Mobile robot control system based on PC104 and network-driven motors. *Robot*, Vol. 27, No. 4 (July 2005) 336-340, ISSN
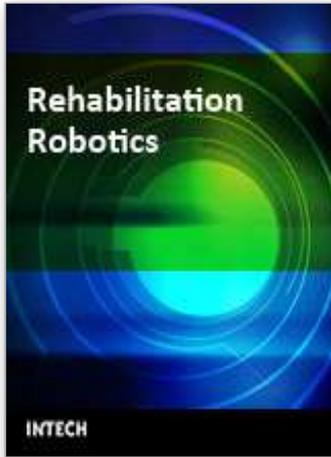
Rehabilitation Robotics

Edited by Sashi S Kommu

The coupling of several areas of the medical field with recent advances in robotic systems has seen a paradigm shift in our approach to selected sectors of medical care, especially over the last decade. Rehabilitation medicine is one such area. The development of advanced robotic systems has ushered with it an exponential number of trials and experiments aimed at optimising restoration of quality of life to those who are physically debilitated. Despite these developments, there remains a paucity in the presentation of these advances in the form of a comprehensive tool. This book was written to present the most recent advances in rehabilitation robotics known to date from the perspective of some of the leading experts in the field and presents an interesting array of developments put into 33 comprehensive chapters. The chapters are presented in a way that the reader will get a seamless impression of the current concepts of optimal modes of both experimental and ap- plicable roles of robotic devices.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Zhang Fuxiang (2007). An Embedded Control Platform of a Continuous Passive Motion Machine for Injured Fingers, Rehabilitation Robotics, Sashi S Kommu (Ed.), ISBN: 978-3-902613-04-2, InTech, Available from: http://www.intechopen.com/books/rehabilitation_robotics/an_embedded_control_platform_of_a_continuous_passive_motion_machine_for_injured_fingers

INTECH

open science | open minds