J. Biomedical Science and Engineering, 2015, 8, 643-652
Published Online September 2015 in SciRes. http://www.scirp.org/journal/jbise
http://dx.doi.org/10.4236/jbise.2015.89060

Alternative Approach for Studying Contractility of the Isolated Muscle Preparations in Real-Time Mode

Alexander Balakin, Ruslan Lisin, Alexey Smoluk*, Yuri Protsenko

Institute of Immunology and Physiology of the Ural Branch of the RAS, Ekaterinburg, Russia
Email: *azuredragon@yandex.ru

Received 21 July 2015; accepted 13 September 2015; published 16 September 2015

Abstract

The real-time control over the observed parameters is known to be necessary during the experiments with biological objects. The use of approaches, such as real-time operating systems, often requires significant material or human resources (operating time, the special administration and programming skills and etc.). In this paper we propose an alternative approach for the implementation of real-time mode in the experiments on muscle preparations (papillary muscles, trabeculae, walls of vessels) based on an external input-output cards with built-in analog-to-digital/digital-to-analog converters and signal processor. As a result, we developed a hardware-software complex for studying the mechanical properties of biological materials (muscles) in real-time mode. This solution has a convenient and easy-to-use user interface, and requires a minimum of computing resources and minimal cost of electricity.

Keywords

Software, Muscle Tissue, Real-Time

1. Introduction

The analysis devoted to the muscle response caused by length changes continues to be one of the most actual issues concerning the regulation of skeletal and heart muscles contractility, as well as their mechanical properties in the passive state [1]. In this case, the apparent homogeneous mechanical activity at the level of whole organ (heart) is determined by the spatio-temporal coordination of inhomogeneous cardiomyocytes in different layers of heart wall [2]. Independently of assigned task, the mechanical measurements on striated muscle require monitoring and control of the varying parameters such as muscle length, strain (as a function of the length),

*Corresponding author.

mechanical rigidity, etc. [3]. The use of technology with analog feedback allowing the control on the experimental process and data collection is one of the best methods of investigation [4] [5]. Currently, digital-to-analog and analog-to-digital technology is mostly developed. In real-time mode, the data processing and signalling rates are getting closer to analog one. Due to the fact that the researcher has a large number of control parameters (electrical, mechanical, thermal and etc.) there is a need for a digital-analog software and hardware complexes to carry out experiments in real-time mode [3] [4] [6]-[8]. Such an approach has several advantages: "on-the-fly" processing data; the use of a variety of control algorithms; the ability to synchronize data transmission and external devices control; ease of data storage. However, there are some limitations. One of the most significant limitations is the problem of choosing the time scale which provides the required accuracy and completeness of data measurement. Time scale should not only allow receiving data on the changes in the properties of the examined object, but also enable control of its properties during the experiment. The real-time mode is most adequate for solving the problem but it also has some complications in implementation. This mode implies measurement rendering and process control at rates when the delays caused by the transmission and processing information do not disturb process itself and do not affect the measurement. It is typical to use "soft" real-time mode for biomechanical experiments on striated muscles. In this case, the response time of the experimental complex admits a specific error, which does not lead to a distortion of measurement results.

Currently, real-time operating systems (OS) based on *MS Windows* or *Unix* are widely used in addition to the experimental complexes, which are based on the experimental platform (e.g. *LabView*, *PowerLab* and etc.). In such systems the real-time mode is achieved by renunciation of certain functionality of original OS, as well as by optimizing the system processes (such as data transmission and processing). Such systems have several advantages, the main of which is the maximum use of available resources of central processing unit (CPU), graphics processing unit (GPU) and random access memory (RAM). However, this advantage results in significant restrictions. Because these systems are generally used to solve very specific tasks there is a problem of flexibility and scalability of such systems due to increasing amounts of the observed parameters or increase in complexity of task under consideration. As a result, the original system would require significant redesign and possibly additional equipment. Increased requirements for the experimental procedure (e.g. measurement accuracy, number of objects and additional control of parameters) increase the requirements for experimental complex. This leads to the obvious increase in maintenance costs and power consumption (which is needed, for example, for cooling equipment). Additional space may also be needed for the new computational equipment. In terms of economy of labor, time and material resources it is necessary to minimize the costs of such operations.

Most of the solutions that implement the use of real-time mode are proprietary. In recent years, OS based on *Unix* have been used more often. The *Unix*-based OS allow not only using of real-time mode, but also configuring the system processes [9]-[12]. On the other hand, the implementation of hardware and software experimental complexes, which are based on similar OS, requires the special skills in the administration and programming specific to the software (for example, the development of drivers, algorithms of interaction, etc.). In addition, the solutions built on the OS with support of real-time mode are not flexible (*Windows* solutions are not suitable for *Unix* and vice versa; there is uncertainty in choosing the IDE to visualize processes in various *Unix*-like OS). As a result, the development of the final solution is a very labor-intensive process (sometimes with the assistance of third-party developers, which influences the cost of the resulting hardware and software complex and time of its implementation).

On the basis of the above-stated reasons the method of carrying out a biomechanical experiments, which would allow recording all the necessary data at the lowest cost to the infrastructure of the experimental facility and power consumption, will be the most promising. Such a solution should have the flexibility in terms of custom settings and variability in the choice of current tasks without a total change of experimental equipment. It is desirable that the source code (computer program) of the resulting solution is freely available in the scientific and IT-community for well-timed optimization during continuous development of new technologies.

In this paper, we propose a possible solution to the above problems. The proposed experimental complex was developed on the basis of the external input-output (I/O) cards with a signal processor, allowing carrying out biomechanical experiments on muscle preparations.

2. Methods

2.1. I/O Board with a Signal Processor (Renunciation of the Real-Time OS)

Previously, for studying the mechanical activity of isolated myocardial preparations in real-time mode we used a hardware-software complex built on the basis of *HyperKernel* subsystem for *Windows* [13]. The control algorithms for length changes of two myocardial preparations contracted in isolation and during their mechanical interaction (by parallel and series connection as the experimental model of whole heart wall segments interaction) were developed on the basis of this complex [13]. However, *HyperKernel* subsystem has a number of limitations: *Windows* version should be no newer than *XP Professional Service Pack* 2; the system kernel is limited by *Standard PC* or *Standard PC* with *ACPI*; there is no support *Hyper-threading* technology. In addition, we used a demo version of this system where the processing time was limited by 30 minutes. These limitations did not allow us to use CPU with a frequency of more than 3 GHz. Also, *HyperKernel* subsystem is no longer supported by its developers. This resulted in some difficulties in the use of *HyperKernel* with modern a computer.

To avoid this problem we decided to take I/O board of analog and digital signals as the basis of software and experimental complex in which the data processing and generation of control signals occur in real-time mode within the board itself. This paper presents the experience of the application of such a module (*L-Card L*-502, http://www.lcard.ru) being combined with experimental complex for biomechanical measurements. This approach significantly enhances the ability of our hardware and software experimental complex. Our complex allows someone to carry out experimental tests with two isolated myocardial preparations in real-time mode simultaneously. A special feature of *L-Card L*-502 is that it contains its own signal *Blackfin* processor and built-in memory that can receive and process the data from an analog-to-digital converter (ADC), as well as control linear servo motors through digital-to-analog converters (DAC).

The main characteristics of the module *L*-502 are: signal *Blackfin* processor with a clock speed of 530 MHz, 32 MB RAM, JTAG input (allow someone to use predefined function for signal processing and unit control within the *L*-502). In addition, it is possible to develop and store your own custom functions. The maximum power consumption of the *L*-502 connected to the *PCI Express* slot is only 7.6 W.

2.2. Selection of the Time Discrete Value for Signal Processing Control

Electromechanical (excitation-contraction coupling) processes in the muscles occur in a relatively short period of time. For example, the action potential duration in rat left ventricular cardiomyocytes is about 150 ms, and the duration of the heart cycle is in the range of 300 - 400 ms. Therefore, the processing time and forming a command signal for a given time interval should correspond to the occurring processes.

Any ADC has a nonzero signal conversion time from analog to digital (it depends both on the software and hardware of the medium). Stable ADC conversion time in *L*-502 module is no more than 4 μs per channel. To control the current length of the preparations it is necessary to receive and process length and force signals for the first and the second preparation. In this case, the time spent on the sampling only 4 channels with the conversion of analog signals from the transducers of the experimental complex into digital data, is not less than 20 μs. Moreover, it is necessary to take into account the time spent on processing input signals (analog and digital), external input command (described below), the formation of the output control signal (in this case, for two DAC). Thus, the resulting time required for stable operation of the hardware-software complex, is not less than 50 μs. Comparing the operating speed of the module and the duration of the registered processes, we used discrete time of 100 μs to arrange stable receiving and signal control. Also, this time discrete allows addition sampling of about four analog channels.

Moreover, there exists a possibility of scalability for *L*-502 module at which one can increase the number of modules that are synchronized with respect to each other on the internal or external synchronization source (this synchronization is fully customizable). Such an approach allows expanding the possibilities of the experimental setup due to the fact that the collection, processing and transmission of data occurs independently within each module firmware (without changing the discrete time). It absolutely does not violate the operation of the setup in real-time mode (at the time, as a subsystem *HyperKernel* with discrete control time of 100 μs spends 50 μs on the inquiry channels and processing of the received signal (the action period *HyperKernel* only) and 50 μs is given for a common *Windows* functionality).

2.3. Operating Principle of Hardware and Software Complex

For experimental complex software we designed two programs: the firmware and user application. The first is loaded into the *L*-502 module and operates in real-time mode. The second contains a graphical interface. User application allows controlling the operation of the first program by passing it commands in the form of data sets and works like a common *Windows* or *Unix* program. **Figure 1** shows a block diagram of the interaction between user interface and I/O board during a single control cycle of 100 μs.

The user application via the commands with the control data establishes a connection with the I/O board program through allocated memory buffer. The module program checks the transfer of the command into the module. If the result is positive, commands are processed by corresponding algorithms described in the module firmware. Then the I/O board program takes the data from the ADC and the digital ports gathered within the allocated data buffers (buffer size for the ADC data is automatically set by the module driver, or defined by user). Data from the ADC are processed with firmware using specified algorithms, whereupon it generates control signals for the servo motors (for muscle length changes). Further the processed data are transferred to processed data buffer available for the user application (buffer size for the ADC data is automatically set by the module driver, or defined by user), then the control signals consigned to the servo motors are sent to the DAC. User application checks the availability of processed data buffer from the ADC: if that buffer is available, the data are displayed on the screen. After this step the control cycle ends. Data are saved in unencrypted structured (binary) form in read-only memory of the PC for further processing. All the experimental data processing algorithms were developed in our laboratory. At the same time there is the possibility of development any other custom algorithms.

User application of the experimental complex can be developed using the following solutions: *Microsoft Visual C++*, *Borland C++ Builder*, *Borland Delphi*, *LabView*, *GCC*, etc. The firmware of the module can be assembled both in the *VisualDSP* environment and using freeware *GCC* compiler (using *GNU Toolchain* for *BlackFin*). It is a I/O board program that contains specially developed length servo motor algorithms.

System requirements for using the experimental complex are defined by the minimum requirements for the OS and performed tasks. For example, if a PC running *MS Windows*, the system requirements can be found out at the official site of *MS Windows* (*windows.microsoft.com*). **Figure 2** shows a general interface of the user application. **Figure 3** shows a schematic diagram of the experimental complex to work with muscle preparations.

3. Facilities

Method of studying the mechanical (length-force) interaction between two isolated myocardial preparations in real-time mode during a "contraction-relaxation" cycle was chosen to check the adequacy of the hardware and software experimental complex [13]-[16].

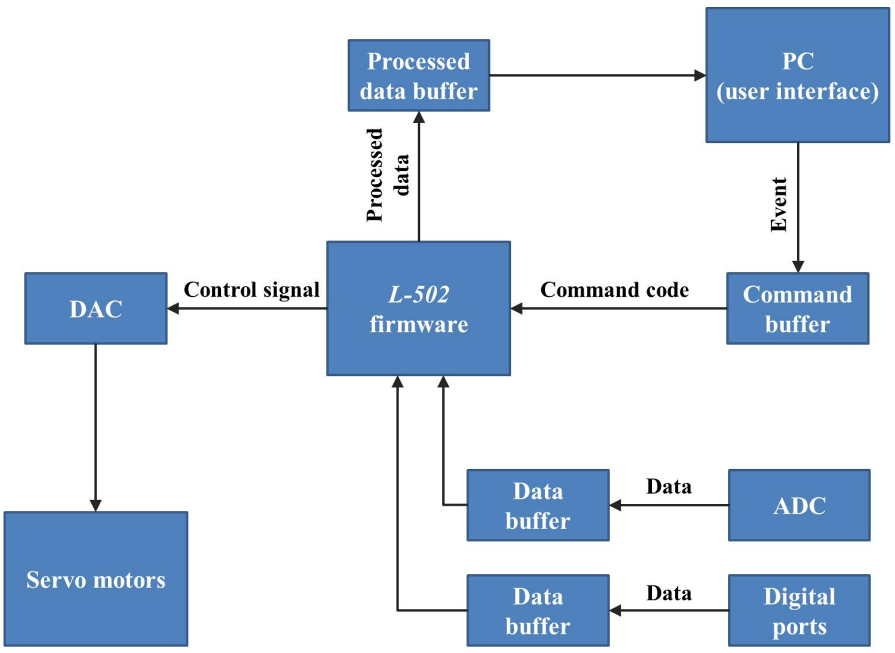
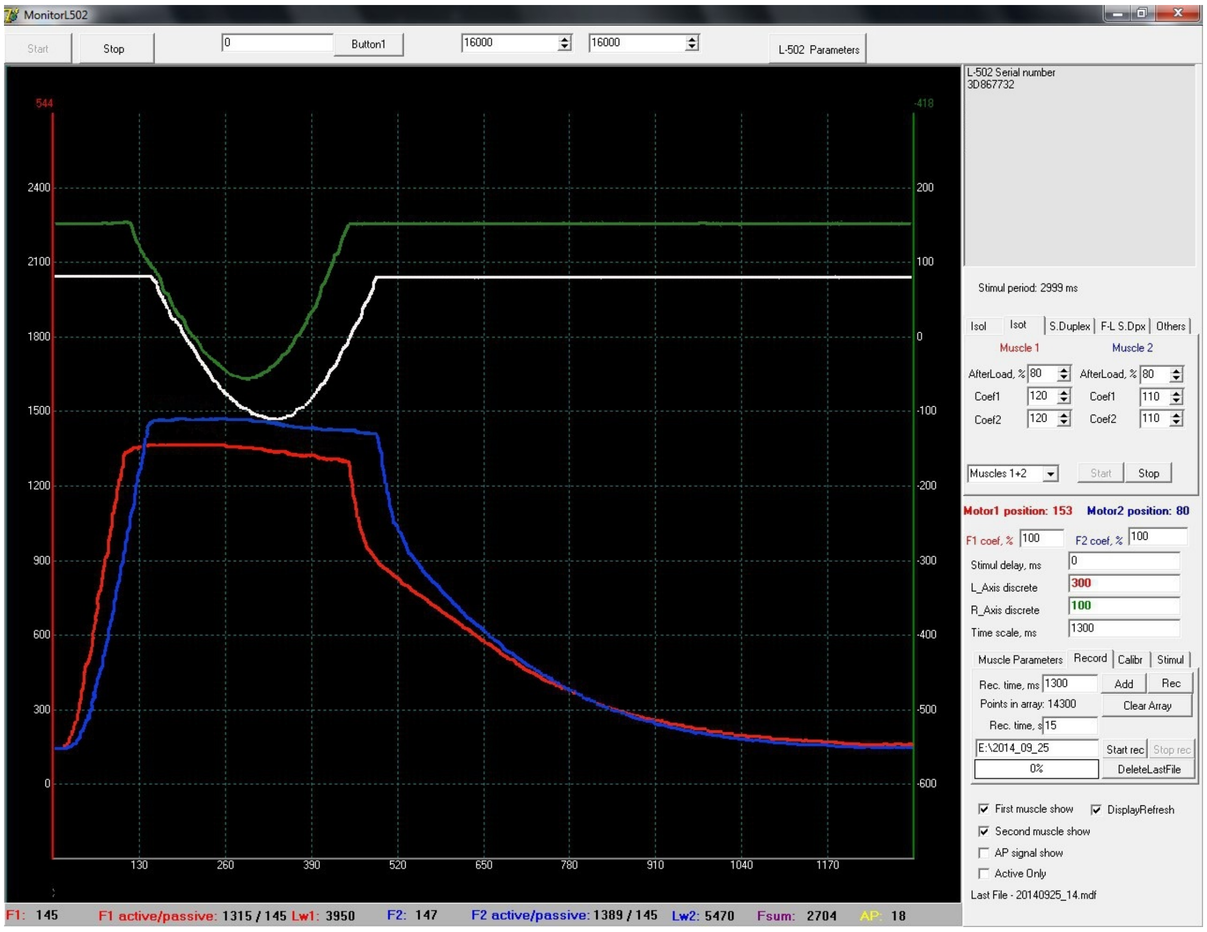


Figure 1. The scheme of data processing between user interface and I/O board during one control cycle.



0

16000

16000

L-502 Serial number
3D867732

Stimul period: 2999 ms

AfterLoad, % 80 | AfterLoad, % 80
Coef1 120 | Coef1 110
Coef2 120 | Coef2 110

Muscles 1+2

Motor1 position: 153 Motor2 position: 80

F1 coef, % 100 F2 coef, % 100

Stimul delay, ms 0
L_Axis discrete 300
R_Axis discrete 100
Time scale, ms 1300

Rec. time, ms 1300
Points in array: 14300
Rec. time, s 15
E:\2014_09_25
0%

☑ First muscle show ☑ DisplayRefresh
☑ Second muscle show
☐ AP signal show
☐ Active Only

Last File - 20140925_14.mdf

F1: 145 F1 active/passive: 1315 / 145 Lw1: 3950 F2: 147 F2 active/passive: 1389 / 145 Lw2: 5470 Fsum: 2704 AP: 18

Figure 2. The user interface. Top two curves represent length and bottom two curves-force of two papillary muscles of rats during the individual isotonic contractions respectively.

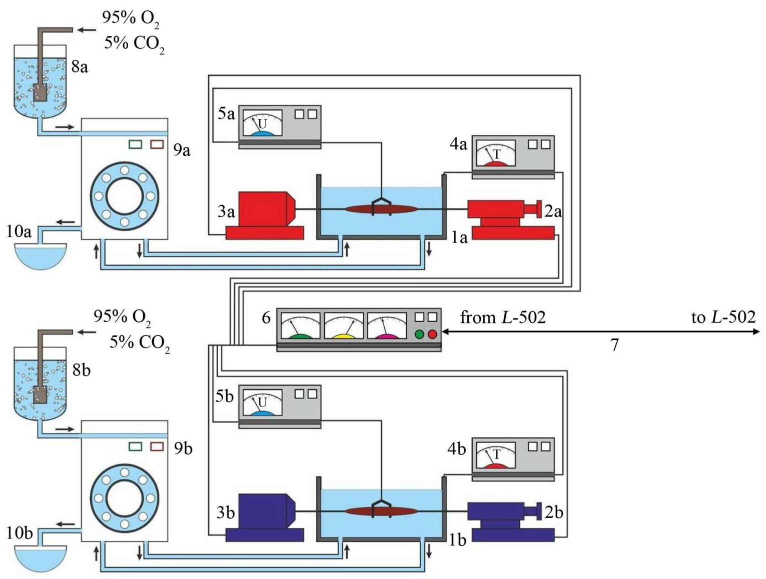


Figure 3. The scheme of 2-channel (a and b) experimental equipment: 1—bath for isolated myocardial preparation, 2-force sensor, 3—servomotor, 4—thermostat, 5—electrostimulator, 6—control unit, 7—communication channels, 8—physiological solution chamber, 9—peristaltic pump, 10—a waste solution tank. The arrows indicate the direction of solution circulation.

Rather small processing and control time discrete (100 μs) practically doesn't distort electromechanical processes of force generation occurring in the muscles and brings them interaction closer to physical interaction.

Control algorithms in software and hardware complex allow implementation of a virtual connection between real muscles. The presence of the two channels of the experimental complex allows exploring myocardial contractility for two preparations at the same time in isolation or in their interaction under different load conditions [13] [16]. During the experiment it is possible to register mechanical properties of muscles, electrical characteristics, as well as dye fluorescence (this functionality will be implemented in near future) in real-time mode. In contrast to previous experimental complex, this new software allows application of longitudinal deformation of arbitrary shape in any phase of the "contraction-relaxation" cycle. Such deformation may be defined both for isolated muscles and muscles coupled in series duplex. Implemented control of electrical stimulators allows simulating the excitation delay between muscles and controlling the order in which they are electrically stimulate. The list of currently implemented modes of our hardware and software is presented below.

1) "Isol" mode. In this mode, the muscle contracts at its constant length, while developing some force (isometric mode). "Isol" mode allows user to adjust the length of the muscle preparations on both channels by changing the position of the servo motors. The value by which it is necessary to change the preparation length is set by the user in the range from 0 to 1000 μm.

2) "Isot" mode. This mode allows setting isotonic contraction regime by holding the force at constant level through changing the length of the preparation due to using feedback loop procedure.

a) On each channel separately (data are displayed from any one channel selected by the user);

b) On both channels simultaneously (channels at the same time are isolated from each other, data are displayed from both channels);

c) On both channels interacting with each other (parallel duplex mode).

When the isotonic mode is on, the software calculates the maximum force developed by the preparation in isometric mode. This value is taken as 100%. Then the preparation is allowed to contract at constant load which is set by user within the range of 0% to 100% of the maximum force. After preparation returns to its initial length, isometric relaxation phase begins.

3) "P. Duplex" mode. In the "parallel duplex" mode the interaction of two muscles, fixed to a mutual base and pulling the mutual load, is simulated (see more details in [13] [16]). The forces of preparations in this approach are added (their sum is equal to the force of the duplex), *i.e.* control setpoint for this mode is performed by following equations:

$$\begin{aligned} F_1 + F_2 &= F_d \\ \Delta L_1 = \Delta L_2 &= \Delta L_d \end{aligned} \tag{1}$$

where $F_1, \Delta L_1$—force developed by the first preparation in duplex and its length change, respectively; $F_2, \Delta L_2$—force developed by the second preparation in duplex and its length change, respectively; $F_d, \Delta L_d$—force developed by duplex and its length change, respectively.

The data from both force transducers come to the input of computational algorithm iteratively. Then data are stored in memory of module. After that the output control signal is generated to change the length of each muscle by the same value to ensure Equality (1). The peak force developed by duplex is taken as 100%. Then duplex is allowed to contract in isotonic mode where each of the preparations shortens by the same value and contracts in auxotonic mode (the force of each element in the duplex are not maintained at a constant level).

4) "S. Duplex" mode. "Serial duplex" mode simulates the interaction of the two muscles "connected" end-to-end [13] [16]. Both isometric and isotonic submodes are implemented in this mode. Control setpoint for this mode is performed by following equations:

$$\begin{aligned} F_1 = F_2 &= F_d \\ L_1 + L_2 &= L_d \end{aligned} \tag{2}$$

where F_1, L_1—force developed by the first preparation in duplex and its length, respectively; F_2, L_2—force developed by the second preparation in duplex and its length, respectively; F_d, L_d—force developed by duplex and its length, respectively.

The data from both force transducers come to the input of computational algorithm iteratively. Then data are stored in memory of module. After that the output control signal is generated to change the length of each muscle by the same value (but opposite in sign) to ensure Equality (2).

a) In isometric submode duplex length is set by user from the preliminary experiments with isolated muscles and actively supported at a constant level during the muscles interaction by active redistribution of their lengths. At any time the force developed by the first element of a serial duplex equals to the force developed by the second element of a serial duplex.

b) In isotonic submode, a maximum force of isometrically contracting serial duplex is calculated; this value is fixed after the mode switching on, and is taken as 100%. The duplex continues to contract in isometric mode until its force is less than afterload while its elements contract in auxotonic mode. Afterwards the duplex starts to contract in isotonic mode under constant afterload and its elements contract in isotonic mode under mutual afterload. At the end of this phase a relaxation of a duplex begins with a further redistribution of the lengths of its elements (duplex isometric mode contraction).

5) "F-L. S. Duplex" mode. This mode provides for tracing performance of the passive preparations in a "serial duplex" mode. During this mode the stimulation of preparations is off and length change for the system as a whole is set. The criterion of stability of the system is that the passive viscoelastic force developed by serial duplex is equal to viscoelastic force developed by each of the elements in the duplex. This mode is placed in a separate category because it provides the study of the viscoelastic characteristics of not only muscle preparations, but also any other biological material.

6) Other modes.

a) "Step". In this mode, step length change of the preparation is assigned. The user can set the amplitude and the rate of length change in the range from 10 μm to 1000 μm at a speed of 100 m/s to 100 μm/hour. In addition, it is possible to set the initiation of step change relatively to the stimulation moment.

b) "Saw". In this mode, saw-like length change is set. The user can set the amplitude and the period of length changes in the same range of speeds and amplitudes as in "step" mode.

c) "Twitch". In this mode periodic U-shaped length change (meander) of the preparation is simulated. The user can set the period and the amplitude of the change. Moreover, it is possible to set the initial pulse time with respect to time of stimulation, as well as to change the duty ratio and the duration of effect.

Another important achievement which is made possible due to the *L*-502 module is a user frequency control of electrical stimulation of preparations. This functionality allows someone to control delay for each of the channels of the experimental setup separately, as well as to control excitation delay between preparation under its interactions. Thus it is possible to simulate the contribution of spatial and temporal heterogeneity of myocardial layers in the pumping function of the heart. It is also possible to improve above mentioned functional modes, as well as to generate totally new modes without any refinement of hardware.

Thus, we developed a biomechanical system that implements the principle of independence between the data reception and interaction functionality and the data storing, processing and imaging functionality. This in contrast to our previous computing system, allows us to increase the speed of signal receiving/transmission by reducing the interruption time, and also to provide additional resources for the user data block without loss of signal quality and improving existing hardware.

As a representative example we have received and processed data in isotonic contractions during the experiment on the living papillary muscle of the right ventricle of rat heart (see **Figure 4**). These data represent the possibility of controlling the length of the preparation in real-time mode so that the value of the force developed by the muscle is at a predetermined level.

4. Discussion

Previously, we have implemented real-time control in our biomechanical measurements on the basis of demo version of *HyperKernel* subsystem for *Windows*. We were encountered with a number of problems when updating obsolete equipment with more modern, as well as limited opportunities for us in the carrying out the experiment on the two muscle preparations simultaneously.

In this paper we propose an alternative implementation of a biological experiment in real-time mode on the basis of the external *L*-502 module. The main technical advantages of the proposed solution are:

1) The real-time mode does not depend on any OS. The whole algorithm of interaction between a personal computer and a control mechanism is realized via the internal program of the external module. This allows taking out some functionality in a parallel flow (e.g. video recording) without loss of system resources.

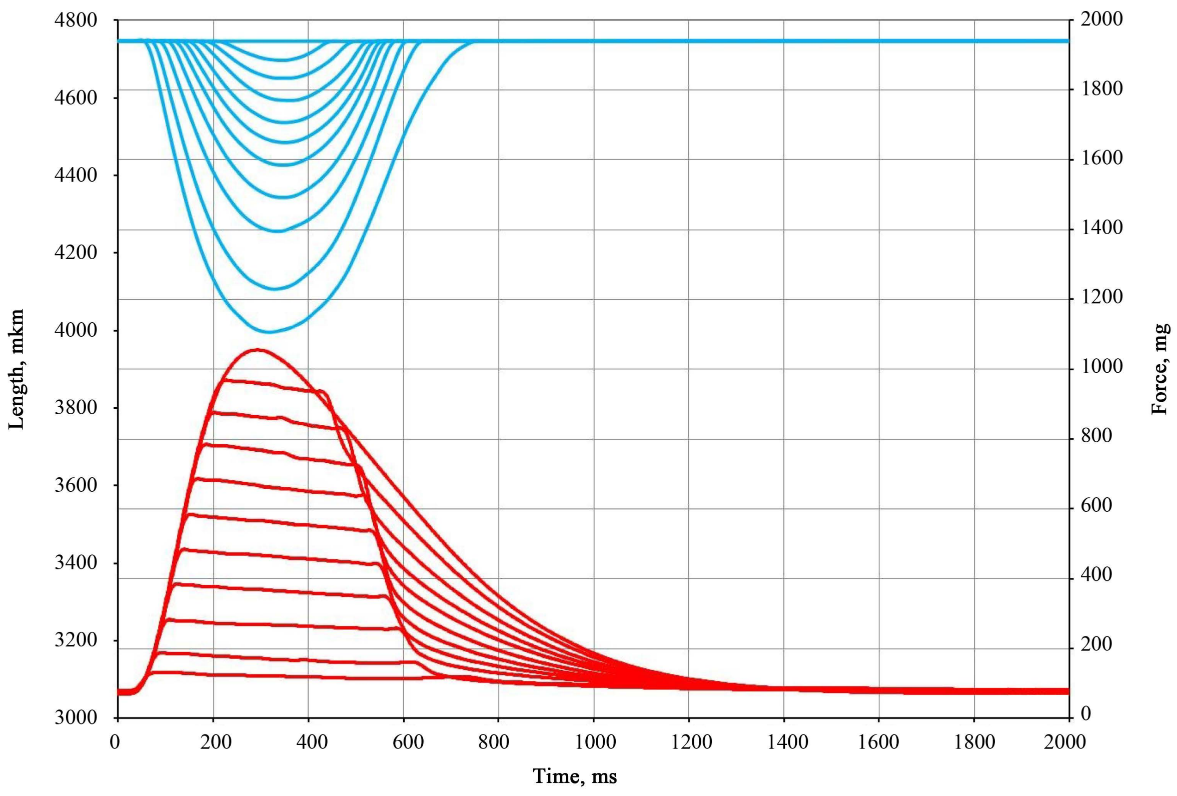


Figure 4. The superposition of isotonic contractions series. The upper curves represent changes in the length of the papillary muscle, lower curves represent changes in the force developed by a preparation.

2) The experimental complex is cross-platform. For operation with an external module it is required only to have a corresponding driver (usually available from the factory in a standard device driver for *Windows* and *Linux*). Moreover, if the user interface is developed on *Delphi*, it is possible to adapt it to *Linux* (via specialized IDE, e.g. *Kylix* or *Lazarus*). There is no need to take into account power of the PC and the capacity of its RAM, because the module uses only its own resources for data processing.

3) There is the possibility to scale the system (in contrast to approaches using real-time OS) by adding new modules synchronized relative to each other on the internal or external clock source. This provides a significant advantage since eliminates the need for fundamental change of the hardware and software complex and expansion of computing power serving the experiment. At the same time scaling absolutely does not violate the real-time mode.

4) The module itself is a very compact device with *PCIe* interface which is compact enough to be used in *MidiTower*. Moreover, the module uses minimal energy and does not require active cooling thus reduceing the costs of energy consumption for the experiment.

5) The implemented complex has an easy-to-use interface which does not require any special programming or administration skills from the scientists.

6) This system is expected to be distributed as free software. Therefore, there is the possibility of improving and developing new functions to the needs of a specific task. Data obtained at the output is stored in unencrypted form, and easily amenable to any statistical processing.

The proposed approach has the following disadvantages. Since resources of external module are severely limited by power of internal CPU and the capacity of internal memory, it is not possible to implement algorithms with complex recursion. But this issue is decided by the system scaling, as well as the use of optimized algorithms. In addition, modern modules cannot be used in the method of hybrid duplex [13], because it requires more powerful CPU and RAM.

Thereby, our hardware and software experimental complex based on the external *L*-502 module provides the unique convenient tool with user-friendly interface for studying mechanical properties of biological materials in real-time mode, which was successfully examined during the biomechanical experiments. This approach is expected to be freely distributed free and recommended for use by other laboratories.

Acknowledgements

We would like to thank Oleg Lookin PhD for assistance with this article. This study was supported by grant RFBR 13-04-00367, 14-04-00085, Program of Scientific the Council of Presidium RAS 2014.

References

[1] Hill, A.V. (1970) First and Last Experiments in Muscle Mechanics. Cambridge University Press, UK.

[2] Markhasin, V.S., Solovyova, O., Katsnelson, L.B., Protsenko, Yu, Kohl, P. and Noble, D. (2003) Mechano-Electric Interactions in Heterogeneous Myocardium: Development of Fundamental Experimental and Theoretical Models. *Progress in Biophysics and Molecular Biology*, **82**, 207-220. http://dx.doi.org/10.1016/S0079-6107(03)00017-8

[3] Campbell, K.S. and Moss, R.L. (2003) SLControl: PC-Based Data Acquisition and Analysis for Muscle Mechanics. *American Journal of Physiology—Heart and Circulatory Physiology*, **285**, H2857-H2864. http://dx.doi.org/10.1152/ajpheart.00295.2003

[4] Peterson, J.N., Hunter, W.C. and Berman, M.R. (1989) Control of Segment Length or Force in Isolated Papillary Muscle: An Adaptive Approach. *American Journal of Physiology—Heart and Circulatory Physiology*, **256**, H1726-H1734.

[5] Iribe, G., Helmes, M. and Kohl, P. (2007) Force-Length Relations in Isolated Intact Cardiomyocytes Subjected to Dynamic Changes in Mechanical Load. *American Journal of Physiology—Heart and Circulatory Physiology*, **292**, H1487-H1497. http://dx.doi.org/10.1152/ajpheart.00909.2006

[6] Sørhus, V., *et al.* (2000) Controlled Auxotonic Twitch in Papillary Muscle: A New Computer-Based Control Approach. *Computers and Biomedical Research*, **33**, 398-415. http://dx.doi.org/10.1006/cbmr.2000.1551

[7] Christini, D.J., *et al.* (1998) Practical Real-Time Computing System for Biomedical Experiment Interface. *Annals of Biomedical Engineering*, **27**, 180-186. http://dx.doi.org/10.1114/1.185

[8] Iravanian, S. and Christini, D.J. (2007) Optical Mapping System with Real-Time Control Capability. *Am J Physiol Heart Circ Physiol*, **293**, H2605-H2611. http://dx.doi.org/10.1152/ajpheart.00588.2007

[9] Dorval, A.D., Christini, D.J. and White, J.A. (2001) Real-Time Linux Dynamic Clamp: A Fast and Flexible Way to Construct Virtual Ion Channels in Living Cells. *Annals of Biomedical Engineering*, **29**, 897-907. http://dx.doi.org/10.1114/1.1408929

[10] Idoux, E. and Mertz, J. (2011) Control of Local Intracellular Calcium Concentration with Dynamic-Clamp Controlled 2-Photon Uncaging. *PLoS One*, **6**, e28685. http://dx.doi.org/10.1371/journal.pone.0028685

[11] Kanu, U., Iravanian, S., Gilmour, R.F. and Christini D.J. (2010) Control of Action Potential Duration Alternans in Canine Ventricular Tissue. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (*EMBC*), Buenos Aires, 31 August-4 September 2010, 1997-2000. http://dx.doi.org/10.1109/iembs.2010.5627828

[12] Lin, R. (2011) Real-time eXperiment Interface (RTXI) User Guide. Boston University, Boston.

[13] Protsenko, Y.L., *et al.* (2005) Hybrid Duplex: A Novel Method to Study the Contractile Function of Heterogeneous Myocardium. *American Journal of Physiology—Heart and Circulatory Physiology*, **289**, H2733-H2746. http://dx.doi.org/10.1152/ajpheart.00306.2005

[14] Smoluk, L.T., Lisin, R.V., Kuznetsov, D.A. and Protsenko, Yu.L. (2012) Viscoelastic Properties of Relaxed Papillary Muscle under Physiological Hypertrophy. *Biophysics*, **57**, 525-529. http://dx.doi.org/10.1134/S0006350912040197

[15] Smoluk, L.T. and Protsenko, Yu.L. (2010) Mechanical Properties of Passive Myocardium: Experiment and Mathematical Model. *Biophysics*, **55**, 796-799. http://dx.doi.org/10.1134/S0006350910050209

[16] Solovyova, O., *et al.* (2014) The Cardiac Muscle Duplex as a Method to Study Myocardial Heterogeneity. *Progress in Biophysics and Molecular Biology*, **115**, 115-128. http://dx.doi.org/10.1016/j.pbiomolbio.2014.07.010

Abbreviations

ADC:	analog-to-digital converter
DAC:	digital-to-analog converter
OS:	operating system
CPU:	central processing unit
GPU:	graphics processing unit
RAM:	random access memory
IDE:	integrated development environment