Force-controlled biomechanical prototype for dental restorations

Marco Silva[1], Marco Esteves[1], Rui Falacho[2], and Luis Roseiro[1, 3]

[1] *Coimbra Polytechnic - ISEC, Coimbra, Portugal*
[2] *University of Coimbra, Faculty of Medicine, Coimbra, Portugal*
[3] *Applied Biomechanics Laboratory - IPC/ISEC, Coimbra, Portugal*
E-mail: msilva@isec.pt

***Abstract* - Teeth restorations are some of the most frequently performed clinical treatments in dentistry. The restoration of a broken or decayed tooth can be done using a direct procedure, that is when a cavity or preparation is immediately filled in situ; or indirect, when the restoration is handled in a laboratory (e.g. crown or veneer), and then bonded to a dental preparation previously performed in the tooth area to be restored. In the restoration process, when cementing agents are used, it becomes essential to control the applied force on the restoration-tooth assembly. A proper force control on the cementation material application improves restoration-tooth longevity. Excessive force can cause damage, and low intensity force will not be enough for the correct seating of the restoration. Without force feedback data, the applied force is dependent on the dentist's sensitivity.**

This work presented the hardware of a new biomechanical prototype device that enables a dentist to monitor the acting force on restoration-tooth assembly procedures. The prototype uses a built-in embedded system for force control and it is expected that this new device can play an important role in clinical environment and can also be assumed as a new teaching tool for dentistry students.

Keywords: Dental restorations, Force control, Embedded Systems, intelligent controlled LEDs, Processing Applications

I. INTRODUCTION

Force control is considered important when performing a restoration insertion into a tooth [1], and until now there are still no commercial devices for this task. The insertion force depends on the sensitivity and knowledge of the dentist, who, even being experienced, never had the opportunity to execute the task within a measurable force procedure. The fear of applying excessive force, that can produce a restoration fracture, leads to insufficient force in this clinical procedure. The result is an incomplete seating of the restoration and a consequent cement thickness greater than desirable [2, 3]. This may lead to an early and rapid degradation of the restoration-tooth interface, infiltration with bacterial colonization and failure of the treatment with the strong possibility of causing loss of the tooth in the short/medium term [4].

In this scope, studies were carried out within the framework of an educational project inserted in a master's degree in mechanical engineering, aiming to create a device for applying an adequate force in the execution of specific tasks in the clinical dentistry field.

At this moment, the prototype under development is going through a functional validation phase. For this purpose, two systems based on HBM strain gauges, for force sensing, were built. The first prototype was interfaced with a PC LabVIEW application through a National Instruments NI 9219a data board. This first approach was detailed in [5].

Fig. 1 depicts the second prototype developed for this project. This subsequent system is herein presented and seeks to achieve low cost and straightforward use. When developing a prototype, time spent on design and functional testing plays an important key role. The hardware components used in this prototype were chosen especially because they enable fast prototyping.

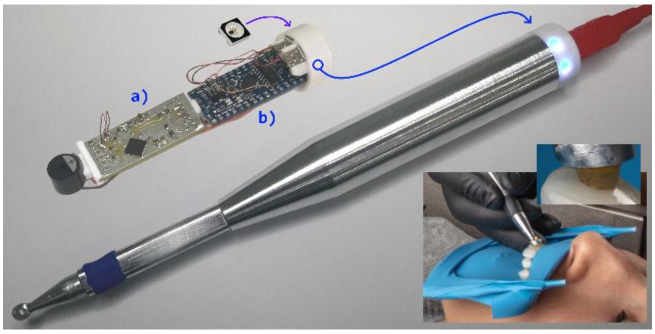


Fig. 1 - Second force-controlled biomechanical prototype for dental restorations.

The main hardware elements of the prototype device are a strain gauge half-bridge with force signal amplification using the MaxLinear's IC XR10910, and an Arduino Nano board integrating an ATmega168 microcontroller. Audible and visual warning information is transmitted to the user by two WS2812B RGB intelligent controlled LEDs and by a common Buzzer. For data recording and device configuration, the system is connected to a PC where a Processing APP (Processing.org) is in execution. This APP allows, as an example, to set up sensing force upper and lower limits, as well as the force acting time interval. Properly configured, this system can assist a dentist by providing real-time force data, as well as recorded force over time with plots for posterior analysis.

II. SYSTEM REQUIREMENTS

The specified base requirements for the prototype were the following:

- The device body should follow the design of a traditional dentistry equipment;
- The device must use a disposable or sanitizable tip/applicator accessory;
- The device force monitoring system must notify the user in real-time, with visual and/or sound indications, if the applied force is inside a predefined range;
- The device must offer the possibility to adjust force low and high threshold indicators value;

- The device must notify the user, in real-time, of the applied force-time and if the time in the range was reached;
- The device must configure the time interval notification value inside a force range;
- The device must provide several force measuring scales;
- Radio devices should be avoided in order to allow easy product certification;
- The device must be able to operate in a connected or unconnected manner using the power of an internal battery.

Design and applied research work has been carried out to achieve these requirements in the development of a medical prototype with force monitoring. These works are described in the next chapters of this document.

III. DEVICE GEOMETRY

The device was developed and optimized with SolidWorks® Software and can be divided into three main parts: the tip, that can be disposable or reusable (Fig. 2 a)); the handle (Fig. 2 c)); and the electronic components enclosure body (Fig. 2 d)). The tip's function is to exert force on the tooth-restoration assembly. The tip has a hole to enable the insertion of a cushion for smooth contact with the tooth.

The device has an instrumented region (Fig. 2 b)) designed for force measuring, where are installed two strain gauges, one on the top and another facing in opposite direction on the bottom. After installation, the strain gauges are covered with a silicon base material for protection and easy cleaning.

The strain gauges are connected in half bridge configuration, enabling to measure the deformations in the specific area.

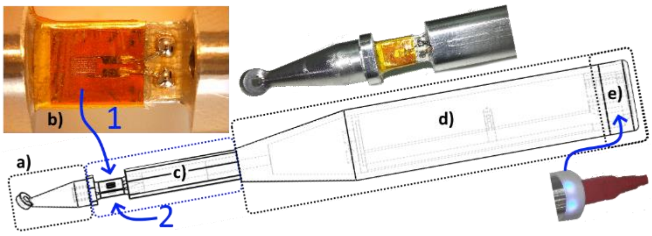


Fig. 2 - 3D Model of the device: a) Tip for designed pad; b) Strain gauges assembly detail; c) Handle to be covered with silicon protection d) Rear support; e) Stopper cap with intelligent LEDs and USB connection.

After the deformation zone there are two holes for connecting the strain gauge terminals to the voltage amplifier circuit that is housed in the rear bracket (Fig. 2 d)). The handle (Fig. 2 c)) has been ergonomically designed to hold the device firmly. The rear bracket has a cylindrical shape to house the electronic circuits needed for system operation, and a USB interface for PC connection (Fig. 2 e)). The mechanical behavior of the device was simulated with finite element method, also using SolidWorks® Software, as described in [5].

IV. DEVICE HARDWARE ARCHITECTURE

The strain gauges' force transducer hardware system architecture is presented in Fig. 3.

In the market there are several programmable analog sensor signal conditioners. Examples are the ZSC31150 and ZSC31014 (idt.com); PGA302, PGA309 and PGA308 (ti.com); MAX1452 and MAX1454 (maximintegrated.com) and the XR10910 (maxlinear.com). There are several differences between them. One of the most important is the differential signal gain, that in some cases is not enough for the target application. Other differences to take care are supply voltage, analog and or digital output, cost, communication interface type, number of inputs, the existence of an internal low dropout regulator to supply the sensor bridge, and the availability of an on-chip temperature sensor that can be used for temperature-compensation (hardware feature of MAX1454).

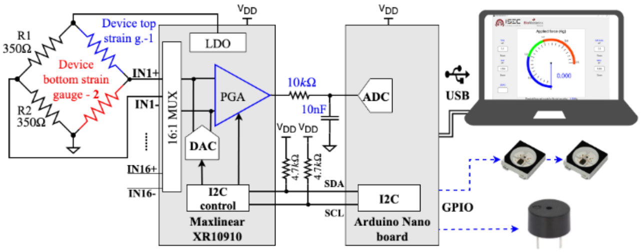


Fig. 3 Force to digital data, device system diagram.

The choice of the XR10910 sensor interface integrated circuit have been based on the fact that it has a simple I^2C communications interface (it is only needed to adjust a set of few parameters), it offers direct analog output, and eight selectable voltage gains from 2V/V to 760V/V. For the chosen mechanical deformation material (Aluminium alloy) and adopted strain gauges (HBM 1-LY13-3/350), 300V/V and 760V/V scales where proven adequate. This type of interface integrated circuits normally has an integrated offset correction DAC to provide digital calibration of the zero offset voltage generated by the strain gauge bridge elements imbalance. The XR10910 has the option of interfacing up to 16 bridges; not a benefit in this case. An absence in this sensor interface IC is a temperature sensor or temperature-compensation method. The zero force offset voltage changes considerably with temperature in this IC. In this prototype device, the development option was based in adjusting this parameter whenever convenient by software commands, through the PC application.

To host a XR10910 chip, a prototype board with all necessary components to implement the sensors' bridge where constructed with a milling process (Fig. 1a)). The software tool used to design this printed circuit board was Autodesk Eagle (autodesk.com/education/free-software/eagle).

When developing a prototype, designing all the needed hardware from scratch is a demanding, time consuming and expensive task. An alternative is to use as much as possible off-the-shelf components. To grant flexibility to a hardware system, the use of a microcontroller is mandatory. For a small size handleable device, a microcontroller development board that fits inside the device is required. From the market available options, the Arduino Nano board can be considered a worthy choice. Arduino Nano is a small, flexible Microcontroller board. The adopted version was equipped with the Atmega168, with components in only one side of the board. All hardware components were fitted on a cylindrical tube with approximately 18*80 mm (Fig. 2 d)).

The Arduino Nano pinout contains 22 digital I/O pins, and 8 analog pins that interface to a 10-bit resolution ADC. The Nano PCB size is 18 x 45 mm (store.arduino.cc/arduino-nano).

The Nano microcontroller ADC is used to perform the XR10910 analog output quantization. The XR10910 analog output ranges from 0 to 3V (it was adopted the default configuration). To scale the ADC voltage input range to the XR10910 output, a 3.3V voltage reference was used. The XR10910 set-up registers are easily configured by I^2C commands.

The development of a software application for the Nano board can be accomplished with the Arduino IDE program (www.arduino.cc/en/Main/Software). The Arduino development environment is a free open-source, cross-platform, simple programming environment that is easy-to-use for beginners, yet flexible for more advanced users. The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries [6].

Available open source libraries are very useful for fast prototyping. The IDE includes a tool to search for shared libraries. Available libraries that were used in this project are the *Adafruit_NeoPixel* and *Wire*. These libraries support a set of code objects that make easy to code control software for a string of WS2812B RGB intelligent controlled LEDs, or to establish I^2C communications, respectively.

To drive a simple 5V buzzer, or produce a pulse width modulation signal, Arduino provides an *analogWrite()*; function that works on the Nano on several I/O pins (3, 9, 10, 11 at 490 Hz; 5, 6 at 980 Hz). When these base frequencies are not suitable for the target application, they can be changed through direct configuration of prescaler registers. Information on how to do this can be found at [7].

V. MICROCONTROLLER APPLICATION

In this case, the simple software application that is required for the target embedded system microcontroller can be developed with the Arduino IDE or a more professional platform like Atmel Studio (microchip.com/mplab/avr-support/atmel-studio-7). For novice programmers an Arduino IDE starting sketch, and associated programing tools, can give a good help in properly structuring code for a microcontroller. Basically, for the target application, the tasks to be coded, must be assigned to synchronous time (by timer) triggered events for force measuring and also perform asynchronous communications data exchange.

The implemented tasks that were coded are the following:

a) Program set-up:

- Include the Adafruit_NeoPixel.h, and Wire.h libraries;
- Create all necessary variables to support code handling and an object assigned to control a string of two WS2812B RGB intelligent controlled LEDs;
- Declare variables to support the process of running a simple low-pass IIR filter for force measurement;
- Define the ADC reference to EXTERNAL to use a 3.3V external quantization window for the strain gauge bridge differential voltage;
- Set and write the color of the RGB LEDS to yellow to signal system startup;
- Configure serial interface to communicate with the PC and with the XR10910;
- Set the gain of the XR10910 to 300V/V.

b) Synchronized tasks at 10 ms:

- Reading of a new force value (10-bit scaled) from the XR10910 and apply the acquired value to the input of a first order low-pass IIR (*Infinite Impulse Response*) filter;
- A process that check if there are new configuration messages from the PC application. These configuration messages enable for example to change the XR10910 amplification value, or all controlled parameters present in the associated PC APP (see below Fig. 5 Processing interface APP.). For simplicity, it was implemented a PC to device communication protocol that uses seven-byte fixed-length messages, with two header bytes. This task involves the use of a state machine that discards malformed messages by checking its values accordingly to the configuration parameters and or with the wrong length. When a good message arrives to the device, the RGB LEDs blink orange and red, and then go back to the normal force representation color.

c) Synchronized tasks at 100 ms:

- At time intervals of 100ms, the 10-bit filtered force value is sent from device to the Processing interface APP, in ASCII format;
- The filtered force value is compared to the threshold force limits, and the RGB LEDS are set to the same color of the PC APP meter, that is blue if the force is insufficient, green if it is inside a defined correct range, and red for over force;
- If the applied force is in the correct range for the configured application, a counter will be set to measure the time interval at which the intended force is applied. In this condition, the buzzer is activated with one beep per second. If the measured force goes over the desired range, the produced warning sound and the time counter stay on. If the measured force goes under the proper range, the beeping and the time counter pauses.

  In case that the pretended time on the proper force is achieved, the warning sound continues on until the measured force goes to the under force range.

  The force time counter is restarted when the time interval is achieved or the system stays in the under force range for more than 5 seconds.

In order to remove slight oscillations in the pointer of the force viewer, improving visual perception, caused by the fact that the device is manually operated, the force values are filtered with a low-pass IIR filter. A low-pass digital filter that can be used on a simple 8-bit processor may be of type IIR which according to the notation in [8], follows equation 1:

$$y[n] = a_0 x[n] + b_1 y[n-1] \tag{1}$$

where: a_0= 1- r; b_1= r; r is a real value between zero and one; n is the current instant; n-1 is the previous instant; r represents the amount of decay between adjacent samples. For example, if r = 0.95, it means that the value in the output signal contemplates 95% of the previous result. The higher the value of r, the slower the decay.

VI. RGB INTELLIGENT LEDS

The WS2812B RGB intelligent controlled LEDs used in this device have a surface mount format with 5*5*1.57mm. A small dedicated PCB board was done to house the LEDs in the device This type of RBG LEDs has an inbuilt controller to process the color they show. Fig. 4 shows that this type of LEDs can be cascaded. To use a LED string, the LEDs only need a common VCC and GND, and a single communications line. A 100nF filter capacitor for power supply decoupling is recommended near each LED. Sending color data for each LED follows a 24-bit transfer protocol that use single NZR communication mode (each LED, follows RGB format, ex: full green - strip.Color(0, 255, 0)). It is possible to send data at speeds of 800kbps. To control a string of LEDs, 24 bits for each LED must be sent in order, with no delay between. Each LED removes its data from the data string sent to the next LED in the chain. More than 280 µs time pauses are used to identify new messages. The great advantages of using this cascading method is that it is only necessary to use one GPIO pin to control a large number of LEDs. Until a power down or a new data command is sent, the LEDs string persist with the last color assigned. The use of a communications library (Adafruit_NeoPixel) makes the programmer task of controlling communication's data format and timing simple.

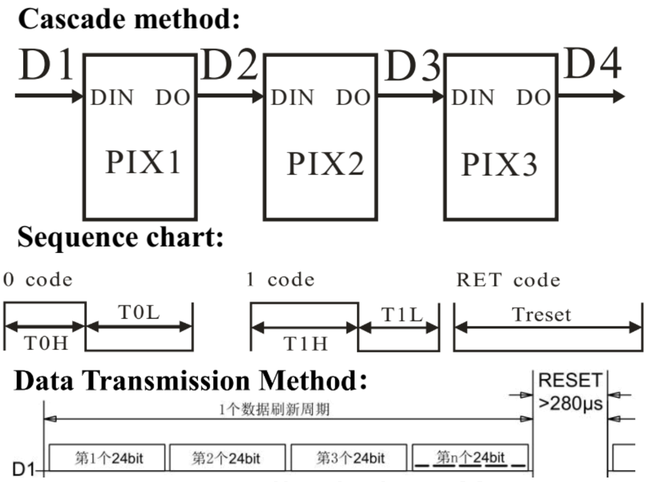


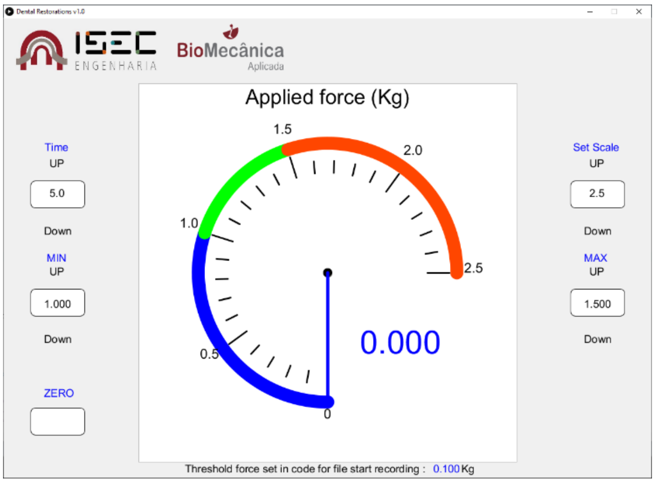
Fig. 4 WS2812B connection and bit coding diagrams [9]

VII. BATTERY POWER OPTION

In the device it was reserved some space below the existing PCBs, to house a small 150mAh 7x15x22mm LiPo battery and the power supply electronic converters for charging and boost the 3.7V of the cell to the 5V supply used in the system. This feature will enable to use the device in standalone mode after being configured with the PC APP.

VIII. PC GRAPHICAL INTERFACE APPLICATION

The PC device interface APP (Fig. 5) was built in the Processing IDE (https://processing.org). The Processing approach has also been applied to electronics through the Arduino project (processing.org/overview/). It is clear the resemblance between the two platforms. Processing includes an open source graphical library and an IDE built for the development of graphical and visual design applications. It aims to teach "non-programmers" the fundamentals of computer programming in a graphic design context. Processing uses the Java language. Being a free and easy-to-use open-source platform for beginners has meant that it is regularly found in PC graphics applications that interface with embedded systems (e.g.: PulseSensor APP: pulsesensor.com/; Snowboard2 APP: kitronyx.com/).



Threshold force set in code for file start recording : 0.100 Kg

Fig. 5 Processing interface APP.

The developed PC APP only uses the included ***serial*** library, for serial data exchange, and the ***meter*** library (https://github.com/BillKujawa/meter.git) developed to display analog reconfigurable meters by Bill Kujawa.

With this simple APP is possible to visualize real-time analog and digital force values. Analog dials offer a visual direct way to check the proper operation of a system. Digital dials offer precise information but is not simple to be interpreted.

All *UP* and *Down* tags near a text box in the APP act as a mouse selection button changing the related variables. Only the mouse release function and location was used to check mouse press events. Every time a "button" is selected it is sent a message from the PC to the device microcontroller, that includes the time interval value under proper acting force, the lower and upper threshold limits, and an end of scale representative value. The force value is scaled in the device through the XR10910 interface with gains of 300V/V and 760V/V. So, the raw 10-bit force value sent from the device is only converted to a force scale in the PC APP. The received raw force values were previously mapped (conversion equation) to a 1kg and 2.5kg end of scale values. The zero offset is also fine adjusted only in the APP.

For teaching purposes, it is very interesting to register the acting force over time. With this purpose, an automatic CSV data files generation was implemented in the APP. A threshold value to start recording force values over time at the rate of 10 values per second is set in an open code variable. When it is detected that the acting force goes back under this threshold for more than a configurable time, the force data that is stored in a dynamic table is written to a CSV file, and the table is cleared. This way, it is very simple to acquire force over-time records without the need of selecting any type of start/stop button. The generated files are easily visualized in charts, where is possible to understand how the procedure is being done, regarding applied force limits and execution time.

As the APP is available in open-source, and as almost all calibration and conversion code is done in the APP, it can be changed to implement new features by the students.

Moreover, Processing apps are platform independent. Then, it is very straightforward to use them on Linux, Mac or Windows seamlessly.

IX. RESULTS AND CONCLUSIONS

In order to get a first validation of the developed device, a set of functional tests have been done in collaboration with a dentist, using artificial restorations.

The performed tests of the device showed good handling and stability, straightforward use, acting as a simple and intuitive force monitoring system with visual and audibly feedback. An added value is that the device has visually and audibly signals directly in the dentist's field of vision, while he performs the restoration.

The next step is to undergo usability tests that will validate all the device functions.

REFERENCES

[1] Cardoso, Ana Sofia de Sousa, Cimentação adesiva: Avaliação da espessura da película em diferentes materiais e técnicas: Estudo Piloto, http://hdl.handle.net/10316/36118, 2016.

[2] Manso, Adriana P and Silva, Nelson RFA and Bonfante, Estevam A and Pegoraro, Thiago A and Dias, Renata A and Carvalho, Ricardo M, "Cements and adhesives for all-ceramic restorations," *Dental Clinics,* vol. 55, no. 2, pp. 311-332, 2011.

[3] Pegoraro, Thiago A and da Silva, Nelson RFA and Carvalho, Ricardo M, "Cements for use in esthetic dentistry," *Dental Clinics of North America,* vol. 51, no. 2, pp. 453--471, 2007.

[4] Simon, James F and Darnell, Laura A, "Considerations for proper selection of dental cements," *Compendium of continuing education in dentistry,* vol. 33, no. 1, pp. 28--30, 2012.

[5] Marco Esteves, Susana Pereira, Rui Falacho, Marco Silva and Luis Roseiro, "Dispositivo Biomecânico para Controlo da Força em Restaurações Dentárias," in *8º Congresso Nacional de Biomecânica, 15 e 16 de fevereiro de 2019, Unhais da Serra,* Covilhã, Portugal, 2019.

[6] Arduino, "Introduction to Arduino," [Online]. Available: https://www.arduino.cc/en/guide/introduction. [Accessed 1 9 2019].

[7] Arduino.cc, "Secrets of Arduino PWM," 2019. [Online]. Available: https://www.arduino.cc/en/Tutorial/SecretsOfArduinoPWM. [Accessed 28 8 2019].

[8] S. W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing," California Technical Publishing, 1997.

[9] WorldSemi, "WS2812B Digital RGB LED," 20 08 2018. [Online]. Available: http://www.world-semi.com/DownLoadFile/111. [Accessed 28 8 2019].