NORMOTHERMIC PERFUSION OF AN ISOLATED LIVER: CONTROL SOFTWARE IMPLEMENTATION

Bachelor thesis in biomedical engineering
September 2014

Author: Lucía Gullón Vega
Tutor: José Luis Jorcano Noval
Director: Juan Francisco del Cañizo López

Universidad Carlos III de Madrid,
Escuela Politécnica Superior

ACKNOWLEDGEMENTS

Me gustaría dar las gracias al Dr. Juan Francisco del Cañizo por su atención cercana, su dedicación y su entera disposición incluso mucho antes de comenzar a ser el director de mi trabajo. A él, y a todas las personas que forman parte del Laboratorio de Circulación Artificial de la Unidad de Medicina y Cirugía Experimental del Hospital Universitario Gregorio Marañón, por el trato amigo e incondicional que me han hecho sentir como en mi propio laboratorio.

INDEX

LIST OF FIGURES

1. ABSTRACT

The current concern about the shortage of available liver donors for transplantation poses different possibilities that help overcoming this problem. New techniques for organ preservation, organ recovery methods of those organs classified as non viable in the first place, or the still under development method of decellularization followed by the recellularization of organs, stand as the most promising solutions nowadays. Along with the primary objective, these approaches also have in common a perfusion step of the organ. This study proposes a newly developed normothermic perfusion machine with a PC control. It provides a user-friendly platform for liver perfusion with the implementation of a software control that accomplishes an autonomous and homogeneous system able to adapt and adjust to the characteristics of the organ. The circuit is constituted by a whole set of devices that are electronically interconnected in order to work one depending on each other, resembling the physiology that interacts with the liver in a real body. These allow an instant measurement of meaningful values and blood parameters of the current state of the system. Results show a stable and constant-parameter control able to work independently of human surveillance during 24 hours while maintaining organ function and viability. It becomes an easier and more accurate system that will improve and make more reliable perfusion experiments in these lines of research that work against the actual lack of liver donors. It offers not only an improved method over the actual ones, but also presents a design able to implement future clinical evaluations into the software environment.

2. INTRODUCTION

2.1. Definition and evolution

In 1813 the first description about the possibility of perfusing an organ was made by Dr. Le Gallois [1], when he enunciated the hypothesis that if the heart could be substituted by a device able to pump arterial blood, any part of the organism could be kept alive for indefinite period of time. The application of this idea in an animal model was implemented later on in 1935 by Dr. Alexis Carrel and Dr. Lindbergh [2] as well as Dr.Demikohv [3].

The pumping effect of the heart provides the proper perfusion of the organism, or/and any part of it. Perfusion comes directly from the verb “to perfuse”. It refers to the introduction of a liquid to the organism in a slow and continuous way. It can be accomplished for an organ, like the liver in this case. The perfused liquid can vary from serum or antibiotics administered to patients, to blood. The normothermic perfusion discussed here will focus mainly on blood.

The system consists of a complete circuit aiming the normothermic perfusion of an isolated liver. It is intended to preserve both the structure and functionality of its cells for periods of time around 3-4 weeks. The design is suitable for big animal models, specifically for pigs, with a view to forward move into human livers. It provides a development aid to overcome actual limitations about hepatic transplants that come from chronic diseases, tumors or hepatic failures.

2.2. Applicability

Organ preservation for transplantation

A current concern in the health field is the demand for transplantation that far exceeds the number of available donor organs. Lately, new approaches from research groups from all over the world are working to overcome this problem.

The conventional technique keeps the newly removed organ in a hypothermic solution for the shortest time possible before transplantation. Organs are both washed and perfused with preservation solution in order to reach a homogeneous low temperature as fast as possible. The applied solution tries to diminish and stop all cellular degradation processes. The importance of the temperature helps to decrease the enzymatic activity [4].

Conventional conservation process is based on the idea that the organ may live longer with the conditions explained. But nowadays lines of research have proved that pretreating the organ with blood and at normal temperature conditions, could enhance even further the post-transplantation function [4]. The perfusion system here becomes the perfect setting to fully develop and study this new line of investigation.

Research studies and organ recovery

One of the reasons for the shortage of available organs for transplant is due to the large percentage of discarded grafts. The vast majority is not considered optimal for the operation and experts fear of inferior survival or biliary complications [5]. Recent advances published in the American Journal of Transplantation [5], have shown a preservation machine working with ex-vivo organs that would be discarded at first sight. They have been working with them at subnormothermic conditions (21ºC), in order to enhance their functionality so to become suitable for transplantation [5].

Maintaining organ viability and keeping its freshness turn to be the main requirements for a successful liver transplantation. It is desired that the system minimizes any related injury that may be caused by the setting conditions. Both temperature and oxygen supply should preserve and recover marginal livers. The perfusion lasts for three hours and the results showed sustainment or improvement of liver function and hepatobiliary parameters postischemia [5].

The discussed perfusion machine constitutes a simpler design than the previous proposed system. The later has into account a higher number of parameters that are accurately monitored real-time. This is the prime cause of using the exposed approach that can adjust the conditions for this subnormothermic perfusion along with several parameters, although it is designed for a normothermic environment.

In addition, a major concern in liver transplantation involves techniques such as reduced liver grafts and extended resections in living donors. The surgical removal of part of the organ produces an hepatic failure known as small-for-size syndrome. It is directly related to the remnant amount of mass of the donor that should not be beyond a threshold point. However, it has been recently hypothesized that the mass is not the exclusive limiting factor, but the hemodynamic parameters of the hepatic circulation. Maneuvers of the hepatic inflow can overcome any large portal perfusion and thus, prevent the development of the syndrome. Flow dynamics can be adjusted to the needs and specifications of any type and size of resection. Leaving the problem to a flow-related and no further mass-related issue [6].

These, and other physiopathological studies can be examine in detailed with the system. It allows changes in pressures, flows and other parameters that can give an idea of the behaviour and responses of the organ when conditions are modified.

Decellularization and recellularization

Tissue engineering proposes a whole-organ decellularization and recellularization processes that provide for three-dimensional matrix scaffolds. Decellularization constitutes the first step involving the perfusion of the organ with specialized detergents. The end result is a completely preserved native scaffold with all previous cells discharged. It maintains the intact vascular network and the appropriate microenvironment that allows for a further perfusion of organ specific cells (recellularization). This second tissue treatment intends to provide new host cells able to locate at the corresponding sites to confer again the proper functionality of the organ. In this manner, any decellularized matrix could be treated for any patient's cells avoiding compatibility/transplant rejection.

In order to fully develop this idea and also, attending to the needs for the decellularization and recellularization processes, a perfusion system can provide the proper development aid for the purpose. It allows both the perfusion of specific detergents (decellularization) and the further cell-rich blood intake (recellularization).

2.3. Background

Gross anatomy of the liver

The liver is the largest organ in the body, contributing about two percent of the total body weight. The basic functional unit is the liver lobule fed mainly through the portal vein and the hepatic artery (Figure 1). About 1050 milliliters of blood flow from the portal vein to the liver sinusoids each minute, and 300 milliliters flow through the hepatic artery [7].

The liver is an organ that extends across the entire abdominal cavity. It is encapsulated by connective tissue. This capsule is further covered and reinforced by the peritoneum that protects the organ and holds it in place within the abdomen [8].

The liver is composed of four distinct lobes. The left and right lobes are the largest ones separated by the falciform ligament. Both small lobes, caudate and quadrate, extend from the posterior side of the right lobe. The caudate wraps around the inferior vena cava and the quadrate around the gallbladder [8].

Hepatic irrigation

The singularity of the liver lies in its main irrigation through the portal vein (around 75%) that transports low-oxygenated venous blood from the digestive tract, pancreas and spleen. On the other hand, the hepatic artery transports 25% of the total irrigation of the liver that constitutes oxygenated blood [9].

Since both intake pathways mix before reaching the hepatocytes, these are never exposed to a complete oxygenated blood. Blood from the portal vein carries nutrients that become in contact with the oxygenated blood from the artery in the sinusoidal capillaries. These ones constitute fenestrated tissues that allow the exchange of substances between the blood and the hepatocytes. The blood entering the organ, as shown in Figure 2, goes through each of the lobules through the central lobular vein and all of them are then collected in the hepatic vein. The venous blood reaches the hepatic veins and finally arrives to the vena cava and returns to the heart to be oxygenated thank to the lungs [9].

Physiology of the liver

The liver takes part in several different roles that involve:

Digestion. The production of bile serves as metabolic substrate to other cells in the organism. It is the main player in the process of digestion [9].

Metabolism. The liver takes carbohydrates, lipids and proteins and transforms them into biologically useful materials.

Detoxification. Enzymes in hepatocytes metabolize toxins into inactive matabolites and keep hormone levels within homeostatic limits [8].

Storage. Essential nutrients, vitamins and minerals from the blood passing through the hepatic vascular system are stored in order to provide constant supply to the tissues in the body [8].

Production. Vital proteins components of blood plasma: prothrombin, fibrinogen and albumin. Other proteins involved in metabolism and transport, glucoproteins and lipoproteins are produced by the liver.

Immunity. Kupffer cells that line in the sinusoids are a type of macrophages that clean large volumes of blood passing through the hepatic portal system very quickly [8].

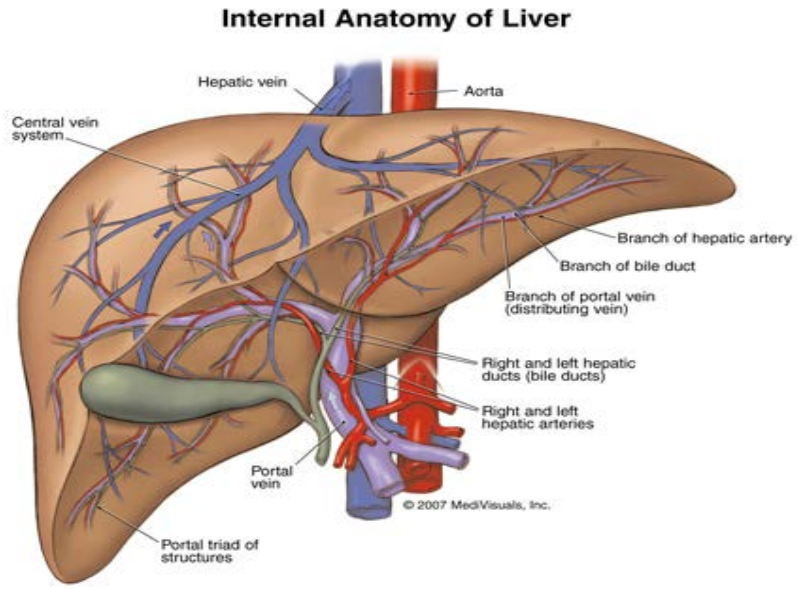


Figure 1. Internal and gross anatomy of the liver

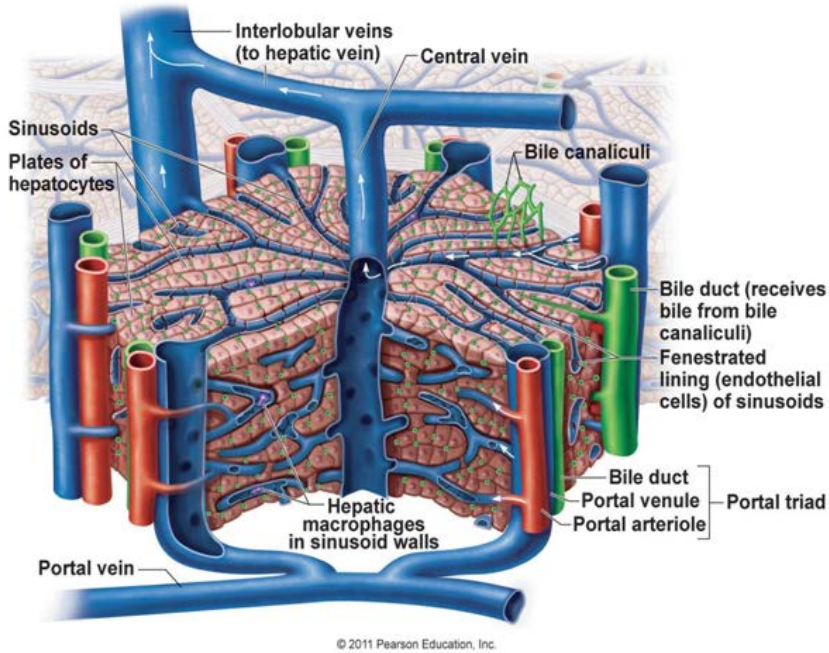


Figure 2. Internal structure of an hepatic lobule

3. OBJECTIVES

The primary objective of the work consists of the implementation of a software for the control mechanism of both pressures and fluxes through the hepatic artery and portal vein. The method is developed in the circuit system for the perfusion of an isolated liver. It encloses:

- Portal vein flow control with pressure limits.
- Hepatic artery flow control with pressure limits.

A secondary objective refers to the implementation of a pH control. This is also integrated within the developed environment of the perfusion circuit. It provides:

- Regulation of the blood pH during the perfusion experiment.

4. MATERIALS AND METHODS

The perfusion system comprises a whole set of devices working together for the homeostasis of the organ. The organ is perfused with blood from the animal and it will be recirculating through the circuit. Since the circuit constitutes a closed system, there is no further intake of new blood and hemolysis becomes a major issue to be aware of. In order to reduce this problem, the circuit accounts with components that are usually encountered in extracorporeal membrane oxygenation systems (ECMO) to the extent possible.

4.1. Circuit components

Figure 3 shows the interconnection of the components and how they assemble to constitute the perfusion system. This one mimics the blood circulation of a real body with the addition of those devices that record continuously the current state of the system.

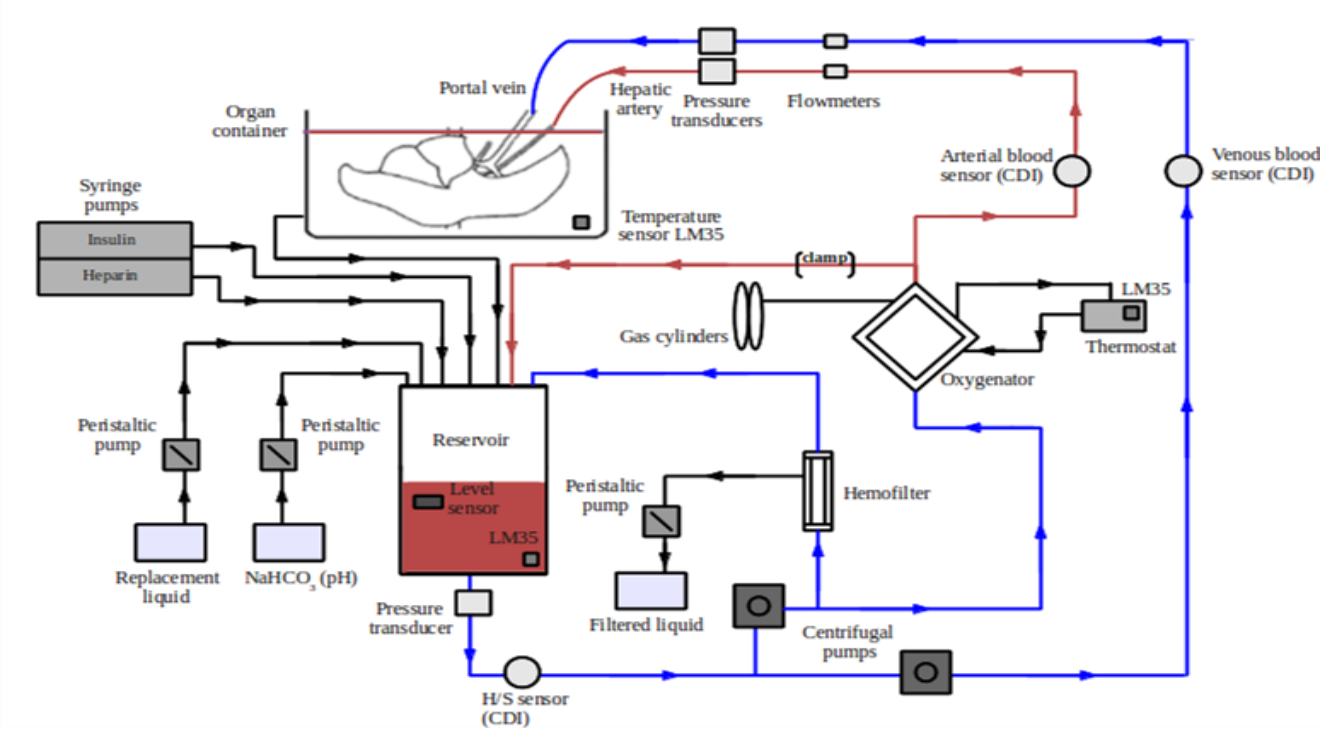


Figure 3. Circuit schematic of the perfusion system

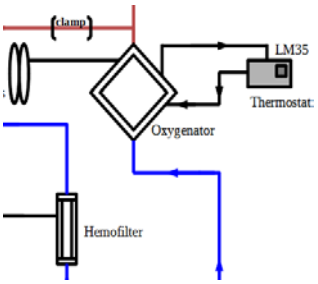
From this sketch, the components are going to be described attending to their functions in the system. The plumbing interconnection for the circuit as much as the electronic development of each of the devices, need to be analyzed in detail. The electronic part refers to the way they work and the communication with the computer that allows the interaction with the user.

Each device requires a specific communication protocol. This refers to the way they accomplish their working mode based on the commands sent and the acquisition of the instant state mode. This coding is always different for each component of the system. This is necessary so that each command is translated for the task and device it was made for.

Arduino boards are used for all the components. This platform interacts with each of the sensors (pressure, flow, level and temperature), actuators (bombs and water heater) and data transfer (CDI500 and XBee). The boards are programmed attending to the specific communication protocol for each of them.

The devices that are controlled through an Arduino board, use a serial transmission. It consists of bits sent over a single wire. This type of transmission specifies a velocity for the communication with the computer. However, in this case before the PC, the communication goes through the Xbee module. This one confers a wireless connection between the perfusion circuit and the PC. This is, each component is provided with this radio frequency module as well as the computer where the user interacts with the system. A defined transmission velocity is required due to this component. At the same time, the Arduino board is connected to the specific board that activates the device or directly to the device itself. This connection is also provided through a serial transmission. This transfer rate will be specified in the program for each of them. The velocity is measured in bauds.

Oxygenator

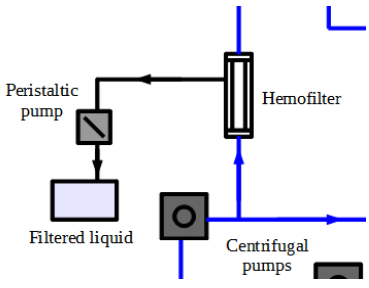


In a real situation, the blood coming to the liver has been circulating and processed by other organs in the body. Lungs are in charge of the exchange of gases: the uptake of oxygen to the bloodstream and the release of carbon dioxide from it. In the bioreactor, this task is carried out by a common pediatric oxygenator QUADROX-iD Pediatric (Maquet, Rastatt, Germany).

The oxigenator is connected to a gas blender that supplies a mixture of oxygen and air controlled by means of two rotameters. The gas cylinders used for the mixture are provided by the hospital[1] facilities, where the experiment is developed.

The oxygenator is also provided with a heat exchanger connection that allows the temperature control of the blood by circulating water through it at the appropriate temperature. Finally, the outcome from this device is then the blood that goes back inside the liver through the hepatic artery.

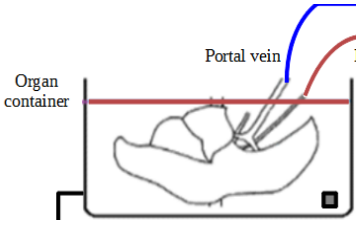
Hemofilter



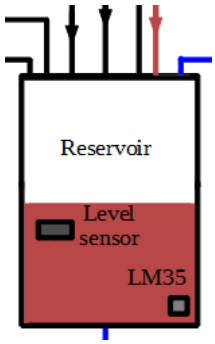
Filtration of blood removes waste products of metabolism found in blood stream. Kidneys provide the essential urinary system that keeps the homeostatic balance that involves electrolytes regulation and urine production. The circuit replaces a real kidney by a Renaflo® II HF Minifilter™ Plus (Minntech Corp., Minneapolis, Minnesota, USA). Part of the blood is filtered and returned back to the reservoir.

[1] Hospital General Universitario Gregorio Marañón de Madrid.

Reservoir and organ container

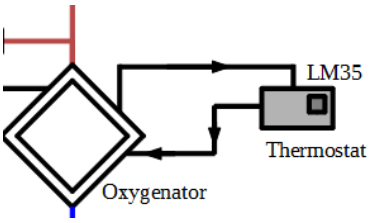


The liver is placed inside a propylene container '500-type Cages+lid' from PANLAB. This is commercialized for rat storage. It provides an approximated volume of 7000cm3 for the liver and the blood surrounding it. For the experiment, the container was drilled with the appropriate holes that provide the recirculation of the overflowing blood where the organ is immersed.



A reservoir is also needed for the storage of the vast majority of blood in the circuit. For this purpose it is selected the VHK 2001 Venous Hardshell Cardiotomy Reservoir (MAQUET, Göteborg Sweden). It is commonly used in open heart surgeries since it is suitable for all operating and perfusion techniques [10]. The container is provided with several entries that allow for the mixture between the substances used as perfusion liquid and an only-one exit. From this part, the blood circulates throughout the system until it goes back to the organ. It becomes a very good option for the storage of the total volume used in the circuit.

Thermostat and temperature sensors



The water is heated in an immersive thermostat Thermotronic II (J.P.SELECTA, Barcelona, Spain). It uses a water container made of methacrylate Clinic-Term as an accessory where the heating coil is immersed. A tube takes water from it and carries it to the oxygenator by means of a small centrifugal pump. Another tube goes in the opposite direction back to the water container.

Temperature is measured at three different points within the circuit. This is, there are three temperature sensors. One of them is immersed in the water container of the thermostat, another one is placed in the reservoir and the third one is immersed in the container where the organ is located. All the three sensors are LM35 temperatures sensors. These can be included both in the Arduino MEGA board along with the level and pressure (spelled out later) or in the one with the thermostat. This is, both Arduino boards are designed so to allow the connection of LM35 sensors for the acquisition of temperatures.

A single box contains the Arduino Leonardo board that includes at the same time the thermostat activation control and the LM35 sensors. This box is exclusively built for these two type of devices. It has its own voltage supply and it is independent from other boards or devices. The relay circuits and the temperature sensors that form part of this, are directly connected to the Arduino board at the specific pins established in the electronics (Fig.4). The program is written for the activation of three relay circuits. However, in our system, only one is used and this one is activated or deactivated. Although the program allows the user to introduce a given percentage, this will be only translated into 100% for activation and 0% for deactivation. The LM35 connected to the board at the analog connections provide the analog temperatures measured and once they arrive, the Arduino software is the one converting them into digital understandable values in ºC. These

calculations are done repeatedly with the counts of a timer. The program only uses one serial port for the Xbee communication at 9600 bauds. Both states of the relay circuits and the temperatures are printed back to the PC. This communication is bidirectional. This is, the computer sends specific commands for each of the functions required:

MTAxxx → Sets the % (xxx) of the relay circuit 1(the only one used for the thermostat).

MTBxxx → Sets the % (xxx) of the relay circuit 2 (not used).

MTCxxx → Sets the % (xxx) of the relay circuit 3 (not used).

MTI → Returns temperatures and relay circuits' percentages (current state).

In the opposite direction, once the command is recognized by the device program, this one send back to the computer the information required attending to a specific coding:

MT*Temp1*Temp2*Temp3*TC1*TC2*TC3\n → Temperatures in ºC of each of the sensors connected (maximum of three) and current percentage states of the relay circuits.

The way the information is written, allows its decoding and user understanding afterwards.

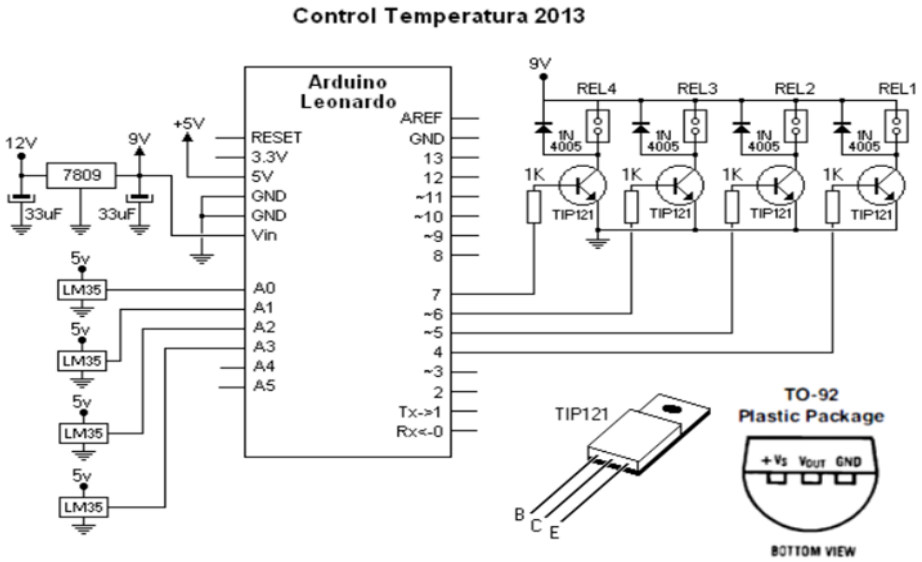
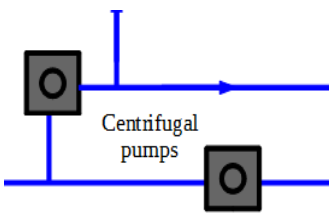


Figure 4. Scheme of the electronics of the thermostat and temperature sensors LM35

Centrifugal pumps



Blood through the body is driven by the heart pumping. This force in the circuit is exerted by two centrifugal pumps commonly used in extracorporeal perfusion circuits. The choice of these types of pumps relies on its pumping effect that helps avoiding hemolysis. From the commercialized head of the emergency kit of ROTAFLOW system

(MAQUET, Göteborg Sweden), the pump is fully developed in the lab (LCA[2]). It provides a continuous flow for the main two entries to the organ.

The interior of the pump encloses a board specific for the motor, an Arduino board with a shield and a LCD[3] interface as shown in Figure 5. The head in the external part is internally connected to the motor. This was analyzed in order to know what is the voltage needed to provide the speed rate selected. The motor of the pump is a brushless electric motor DC driven. It is directly connected to the board that controls the voltage supplied. The incoming power supply is connected to both the main board and the Arduino complex. This one refers to the combination of an Arduino Uno board and an Arduino Prototyping Shield where the parts that go on it can be soldered. This Arduino complex is at the same time connected to the board of the pump and to the pump itself. These two connections provide the interaction between the PC and the pump with a feedback control mechanism. The commands are sent via the board. The voltage provided activates the motor in order to achieve the set value selected by the user. At the same time, an internal feedback control checks the current rpm through the interaction of the Arduino complex with the motor through a dynamo. Another feedback control for the pump rate will be in charge of the software and it involves the difference of rpm related to the communication between the device and the computer.

The pump is also provided with a LCD that allows a rapid manipulation by the user. Three buttons provide menu selection and speed rate modification (increasing o decreasing the selected rpm). It shows the menu selection (setrpm, working mode or stopping mode) when the pump is controlled manually and both the set value and the current rpm. Alternatively, when it is being controlled from the PC the state will be “PC control”. As much as these three buttons, the LCD is also linked to the Arduino complex. The whole device is provided with the connection for the external voltage supply and an ON/OFF button.

The two centrifugal pumps' software program for the Arduino boards are designed in the same way. At the beginning, each of the pins in the Arduino board are set for their corresponding position in the LCD attending to the electronic configuration (Fig.6). For this purpose, Arduino is provided with a library that allows an easy programming. After this, the remaining pins used for buttons or outcoming data are also specified. This is, both the pump and the display are directly connected to the Arduino board. The serial port communication is set at a rate of 9600 bauds. And that is the channel where the incoming string from the PC comes. Once the program checks that the heading is one of those corresponding to the communication protocol, it performs one or another task depending on it. The options could be stopping the pump, setting a specific rpm or sending back the status of the pump. The commands that ask for these functions are the following:

MB1P → Stops the pump.

MB1Vxx → Sets the pump speed (rpm/100).

MB1I → Returns the state of the pump.

[2] Laboratorio de Circulación Artificial (Artificial Circulation Laboratory).
[3] Liquid Crystal Display.

The velocity of the pump is being analyzed repeatedly in time with a counter. The program makes a conversion from the analog value measured by the pump to the digital one sent back to the PC and to the LCD. This information is returned with a specific code:

MB1*setrpm*Vueltas\n → Desired velocity and the actual one in rpm.

The part for the manual control also needs to be programmed. Each of the pumps is provided with three buttons that allow the user to change the mode state and to increase or decrease the velocity and all this is shown in the LCD (set value and the current value for the rpm).

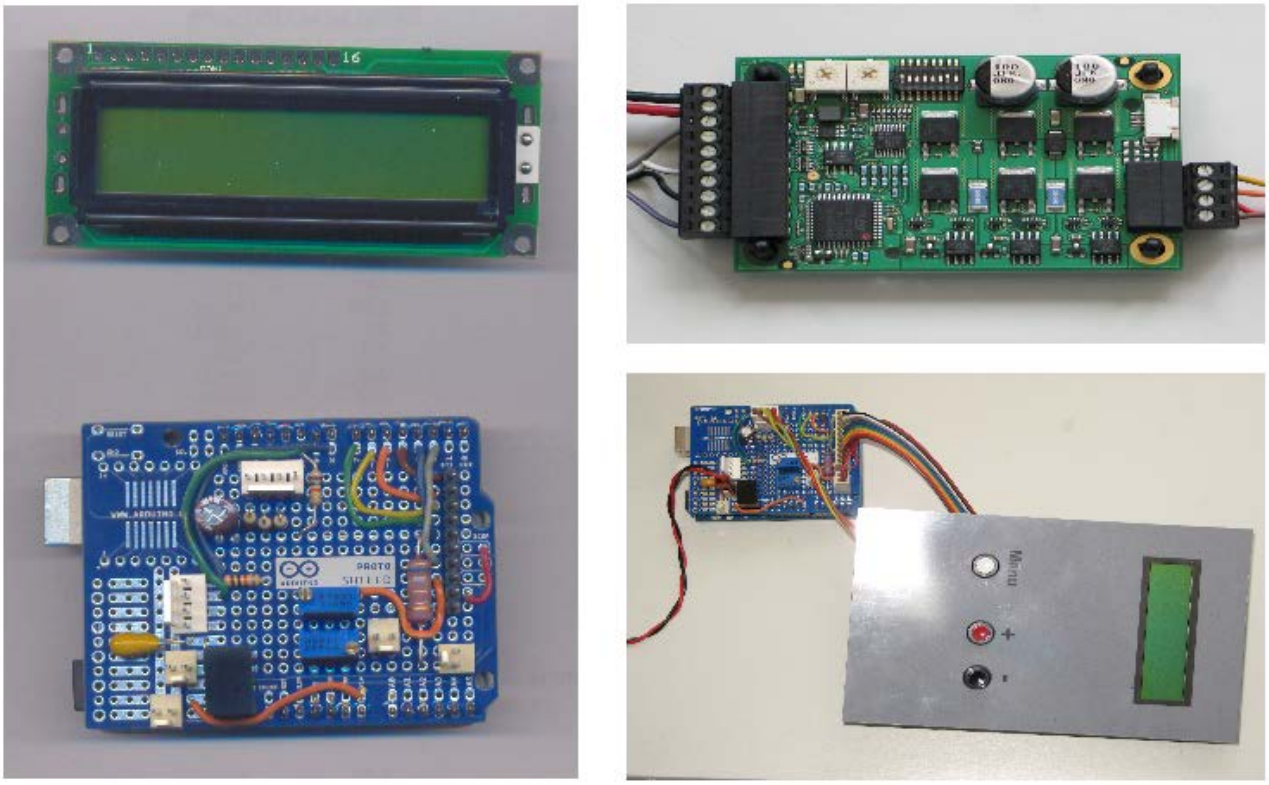


Figure 5. Electronics and circuitry of the ROTAFLOW pump

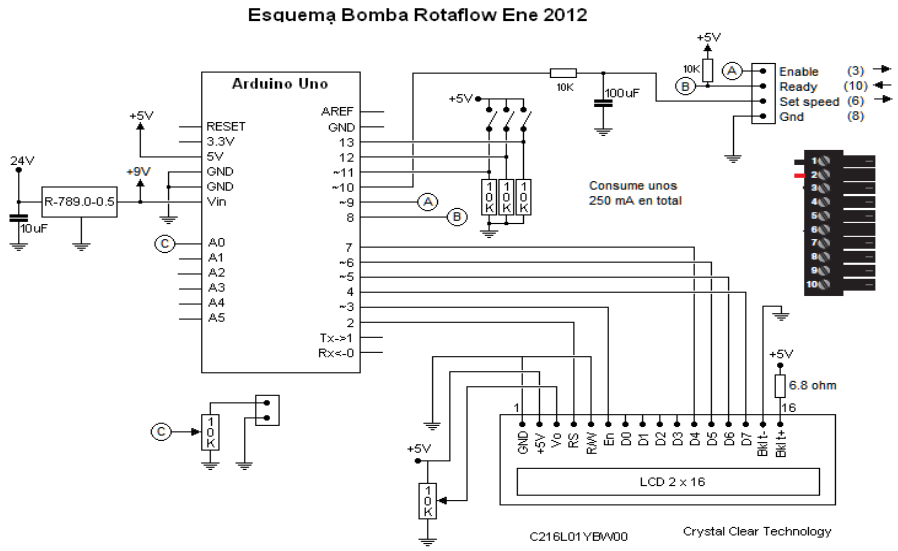
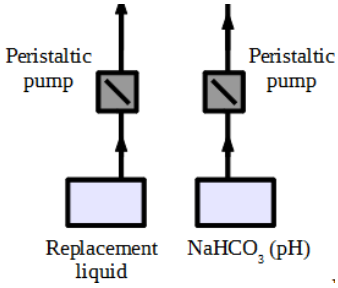


Figure 6. Scheme of the electronics of a ROTAFLOW pump

Peristaltic pumps

In contrast with the previous pumps, filtration is collected using an easy tube loading peristaltic pump with 3-roller rotor from VerderFLEX® EZ OEM pump (Verder, Basildon, UK). As with the previous pumps, this one is connected to a speed control circuit developed in the lab, that provides variable voltage-speed capability. The working mechanism consists of a successive "pinching" of the tube at three locations as shown in Figure 7. The continuous pumping behaviour imitates peristalsis[4], which is the effect that better fits for the purpose of the experiment.



The system uses three of these pumps for filtration, replacement and pH control. Since there is a continuous filtration of blood, the total volume in the circuit decreases. This demands a control that can provide an increment of perfusion solution when required (replacement). This one is rerun with saline serum or glucose solution depending on the organ needs. These two pumps along with the third one for the pH, work independently and they are controlled from the PC-based interface. Their control provides changes in the speed rate in rpm. Each of the devices has an Arduino board incorporated and its software with the corresponding communication protocol that allows processing the commands for the pump speed and gives back its actual state.

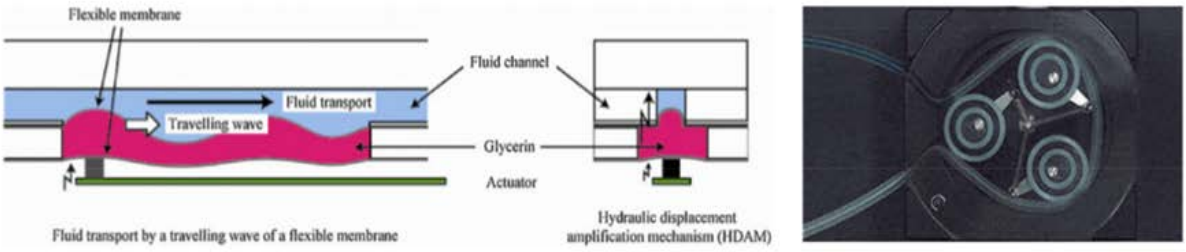


Figure 7. Peristaltic pump's mechanism

Peristaltic pumps are programmed in a similar way as previous devices. First of all, the necessary pins are established according to the electronic connections. A single communication channel is set at a rate of 9600 bauds for the Xbee module. The script analyzes the incoming string and checks that it is an understandable one. There are many options for the VerderFlex pumps. They can be commanded to stop, start at specific velocity, or to be controlled from the PC. The commands for each of these tasks (here specified for pump number 1):

Mb1C → Sets the pump for PC control.

Mb1P → Stops the pump.

Mb1Vxxx → Sets the rpm for the pump.

Mb1I → Returns the current state of the pump.

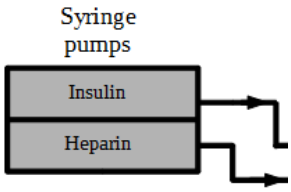
[4] The radially symmetric wave-like propagation of muscle contraction in the digestive tract.

The PC control is done through one of the pumps but allows a user control of the three pumps. The sending commands also carry information about the specific pump the user is referring to. Every time the incoming string is validated and the information that the command required is obtained, this is printed back to the PC as follows:

Mp1*ControlPC*Cuentas\n → State of the PC control (ON/OFF) and rpm of the pump.

The program controls continuously whether the pumps are being controlled from the PC or not and from it, the following commands are decoded. The actual state of the pumps, this is, their velocity, is obtained from the analog value read in the pin. This value is then converted into a digital one. This procedure is done repeatedly and for each of the pumps the user is working with.

Syringe pumps

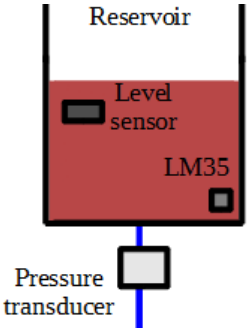


During first few experiments, some blood parameters kept under observation and from this, further implementations were added to the system. Blood showed early stages of the coagulation process, specially as time passed. Extracorporeal circuits induce a “whole body inflammatory response” due to contact of blood and cellular elements with the foreign components of the circuit. This complex interplay of systems produces coagulopathy characterized by the activation of processes such as microvascular coagulation, platelet dysfunction, and enhanced fibrinolysis [11]. This effect could produce irreversible consequences in the organ. Coagulation may start within the tubes of the circuit, avoiding the proper flow of blood coming to the organ. This results in a dysfunction of the liver and consequential damage. Heparin works against this effect thank to its anticoagulant contribution. A continuous and small infusion of heparin is needed to keep the proper behaviour of blood and prevents the system from coagulating.

Alternatively, insulin is related to an excess of glucose also seen during early experiences. Tests showed a high concentration of glucose and it did not diminish during the experiment. Liver cannot decrease large amounts of this parameter by itself because it cannot produce insulin. Hepatocytes absorb much of the glucose entering through the hepatic portal vein, and store it as glycogen [8], but this storage capacity was not enough for the quantities present in the circulation. Two different hypothesis stand for the cause of the high level of glucose. The first one is directly related to the constant administration of parenteral nutrition to the perfused liquid, which is glucose-rich. The other idea comes from a low glucose level state of the animal during surgery. This situation favors the degradation of glycogen stored in the hepatocytes, that is released afterwards within the experiment. Independently of the cause, the level of glucose needs to be controlled, and it is desired to be done in a physiological way as if it were the basal insulin that is controlled. Similarly to the treatment followed with a diabetic patient.

A constant infusion of heparin and insulin became a necessity after first experiences perfusing isolated livers. An external aid was required and sophisticated syringe pumps were selected for continuous and controlled infusions of both substances. Their autonomy also contributes to the PC-based user interface design, especially when the experiment keeps working at night. The Alaris® CC Plus syringe pump (CareFusion, Alcobendas, Spain) is commonly used in

ICU facilities. It provides a large and clear display with an intuitive user interface [12]. It allows a wide range of specialty infusion sets that can be adjusted according to the quantities demanded. They are manually programmed at the beginning of the experiment in contrast with the devices that have an Arduino board integrated for software programming. The pumps have in-line pressure monitoring integrated for early detection of occlusions. Due to its portability, they can be easily integrated in the circuit.



Level sensor and pressure transducers

A VC5510PNOP Capacitive Level Detector (Carlo Gavazzi, Lainate, Italy) is placed on the reservoir of the circuit. The volume of perfusion liquid must remain within specific limits during the perfusion. Constant filtration produces a decrease of this volume that is balanced with the liquid for replenishment. In order to know if the volume is low enough for stopping or activating replenishment, it is needed a level sensor. The device is adjusted at a certain point where the level is considered low. At this point, the signal is sent to one of the peristaltic pumps that activates the process. Level becomes one of the most important parameters to be controlled. It must be continuously tracked and it is desired to commit almost no error. In order to minimize any possible mistake, the control assures the sensing measure with a pressure sensor. This is placed at the exit of the reservoir measuring the pressure exerted by the column of liquid contained in the reservoir. This second idea for the level measurement turned out to work better than the previous capacitive level sensor. The level is recorded more accurately with the pressure transducer. The previous sensor worked well with water but it caused problems when moving to blood. This is the reason why it was finally discarded and not used any more in the system.

The acquisition of flow and pressure is done in a similar way. Pressure transducers are directly connected to the board that controls them. Each Two Channel Invasive Blood Pressure (IBP) OEM board EG 02000 (medlab, Stutensee, Germany) can give the information of two TrueWave Disposable Pressure Transducers (Edwards Lifesciences, Irvine, CA, USA). The boards are at the same time joined together with an Arduino MEGA board that provides the communication with the PC. The software for the Arduino board is specific for the communication protocol of the IBP boards. A single Arduino board integrates at the same time four pressure measures (only three of the entries are being used) from the two IBP boards, temperature from the LM35 sensors and level data. The commands integrated in the software for communications are defined as:

MP1A →Returns the pressure values measured in board A.

MP1B → Returns the pressure values measured in board B.

MP1T → Returns temperatures and level data.

MP1O → Establishes measuring mode in both boards (A and B).

MP1M → Establece el modo simulación en las 2 placas

MP1Z1 → Makes zero at pressure line 1 in board A.

MP1Z2 → Makes zero at pressure line 2 in board A.

MP1Z3 → Makes zero at pressure line 1 in board B.

MP1Z4 → Makes zero at pressure line 2 in board B.

MP1Z5 → Makes zero at four pressure lines.

MP1S → Returns the actual state of both boards.

MP1P0 → Stops instantaneous pressure measures.

MP1P1 → Sends the data about the instantaneous pressures.

MP1V0 → Acquisition rate 50/s.

MP1V1 → Acquisition rate 100/s.

MP1V2 → Acquisition rate 150/s.

Another Arduino Mega board is connected at the same time to two IBP boards for pressure acquisition, and to another board that controls temperatures from the connection of LM35 sensors and the level sensor. This means that the .ino program that controls this whole framework is provided with three serial ports. One of them is for the temperature and level, and for the Xbee interaction in order to communicate with the computer. The other two are used for the data extraction from each of the IBP boards. These wires are set at a rate of 9600 bauds. The first part of the script is focused on the temperature acquisition. Each LM35 is connected to a pin and so it is the capacitive level transducer, and from the analog information they record, this is converted into digital and sent back to the computer when this one asks for the information. The second part is the one dedicated to the pressures. There exist several different commands. There are two pressure boards and each of them presents two transducers connected. This means that this software is working with a maximum of four different pressure values. However, it has been already explained that the system is only using three of them. The ones corresponding to the entries of the organ and a third one to assure the level control of the reservoir. The IBP boards provide a wide range of information about the state of both the transducers and its measures, and about the IBP boards themselves. The program decodes the incoming string based on the coding protocol from IBP OEM board EG 02000 and splits it in different parts according to the commands that are being received. The information coming to the PC differs depending on the data contained:

MP1x*PS1*PM1*PD1*PS2*PM2*PD2*FRC\n → Pressure values measured in each board.

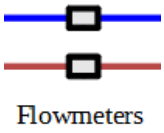
MP1T*Temp1*Temp2*Temp3*Nivel\n → Temperatures measured by the LM35 and level data.

MP1S*Stat1*Stat2*Stat3*Stat4\n → Current state of both boards.

MP1W1*Pres1*Pres2\n → Instant pressure measures from board A.

MP1W2*Pres3*Pres4\n → Instant pressure measures from board B.

Flowmeters



The data for the flow is gathered also using a combination between a transducer and a board. In this case, the OEM Ultrasonic Flow Measuring System DIGIFLOW-EXT1 (Emtec, Gennevilliers, France) with an Ultrasonic Clamp-On Transducer (Emtec, Gennevilliers, France). The transducer is indicated for the contactless volumetric measurement of liquid flowing through extracorporeal tubing systems. The measurement principle is the ultrasound transit-time method. The method relies on the difference in transit time between the waves propagating in and against the direction of flow that is influenced by the medium. A piezoceramic crystal (A) is stimulated by a high frequency burst and sends out ultrasound to a second piezoceramic crystal (B), the receiver. These are arranged in a certain angle a. From this information, then both average velocity of the fluid and the inner cross-sectional area of the tube can be obtained. Straightforwardly the instant value for the flow volume in liters per minute is calculated for both hepatic and venous paths.

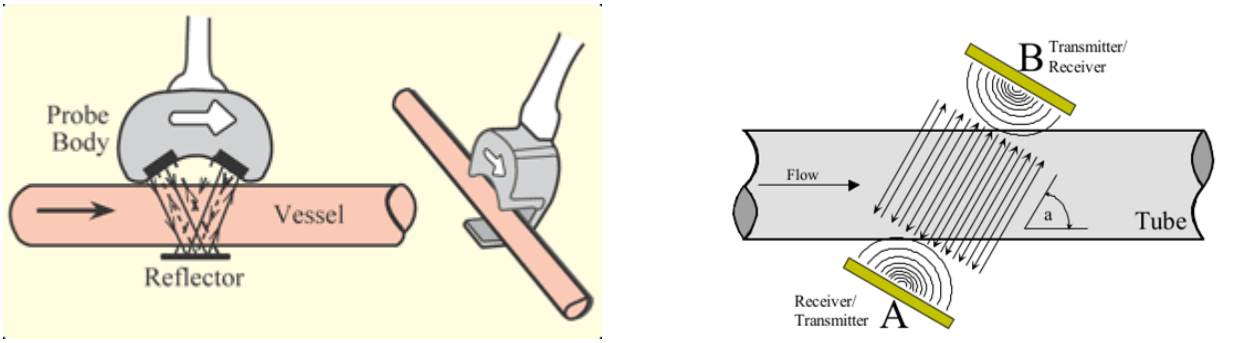


Figure 8. Schematic representation of the measurement principle of flowmeters

The board that controls the flow acquisition is at the same time connected to an Arduino MEGA board that allows the communication with the PC through the Xbee module (wireless). A single board is able to collect the data from just one flow transducer.

The script program for the flowmeters uses three serial ports. Two for each of the transducers (through the boards) and a third one for the Xbee communication. The latter uses a velocity of 9600 bauds while the others communicate at a rate of 38400 bauds. In these two channels, the information from the sensors come and it is analyzed so to be validated. This is, the incoming string must be of a specific length and only then, it will be saved for further delivery to the PC. This second step is done by the Xbee. It first makes sure that the incoming command sent by the user is the correct one, and then the program is able to send back the data it was asked for. The commands can vary between:

MF1A → Returns the measures from transducer 1.

MF1B → Returns the measures from transducer 2.

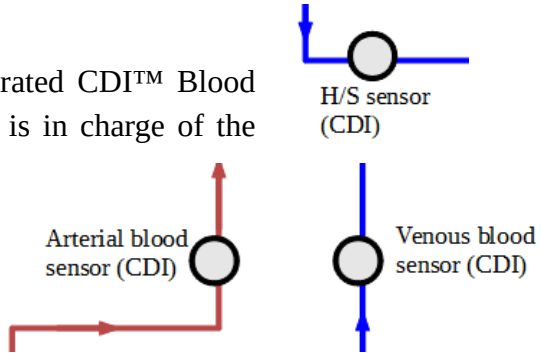
MF1C → Returns the measures from transducer 3.

MF1D → Returns the measures from all the transducers connected.

The string coming back to the PC is decoded as follows:

MF1A*00 E3*0*0*0*0*5503*690*0*1 → Information about the board. Only parameters about the average flow in one second and the temperature of the board are meaningful for the program and used for decoding.

Monitoring device



In addition to these separate devices, a whole integrated CDI™ Blood Parameter Monitoring System 500 (Terumo, Tokyo, Japan) is in charge of the multiple variables acquisition. It constitutes a high technology device that determines every small variation of blood parameters real-time. This data is a determining factor for the feedback control and reaction time. The acquisition of the eleven critical blood gas parameters relies on optical fluorescence and reflectance-based-in-line system [13]. It is composed of three different sensors for venous and arterial blood along with measurements of the hematocrit and blood saturation. The first two are placed right before entering the organ, and the third one is located at the exit of the reservoir before the blood splits into venous and arterial paths. The sensing part of the sensors needs to be changed and calibrated for each perfusion.

The CDI monitor presents two serial ports for communication. One of them, set at a rate of 38400 bauds, is in charge of the storage of the current information coming from the monitor. While a second serial port at 9600 bauds, is the one directly connected to the Xbee. This channel allows the incoming command from the computer, and also provides the current information to be sent back to it. First of all, the program checks that the command sent from the PC is the one asking for the information (CDI), and once this is validated, the data already stored is sent back to the computer for visualization in which the string has the same command as heading. In this case, only one command is used for the communication. The one that asks for the information:

CDI1 → Returns the data shown in the CDI monitor.

And the data is returned with the following code:

CDI1*Hora*pHArterial*PCO2art*PO2art*Temp.art*HCO3*EB*SO2art*K+*V'O2*Q'*pH Venoso*PCO2ven*PO2ven*Temp.ven*HCT*Hgb*SO2 → Specifies each parameter measured in both arterial and venous blood.

Xbee

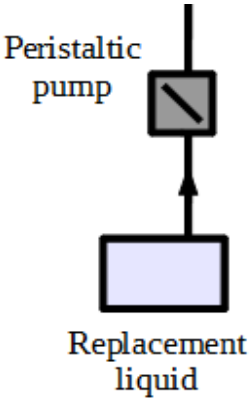
The communication pathway that exists between the PC and each of the devices that compose the system, is made via wireless connection. XBee™ ZNet 2.5 OEM RF Module is in charge of this task. It is connected to the PC through the standard USB communication protocol. These ones are easy-to-use, reliable and cost-effective RF devices that provide a simple method of transmit/receive information in a bidirectional way. This is possible thank to the connection of a

Xbee Wire Antenna module in each of devices already mentioned. These incorporate a small antenna that provides the wireless communication with the main Xbee module connected to the PC.

Perfusion liquid

The organ is perfused with blood from the donor animal. The blood recirculates before the organ is placed in the bioreactor. This is because it needs to be pretreated with subtances that prevent from adverse effects and help for the organ maintenance. Heparin prevents the coagulation process; methylprednisolone avoids antinflammatory effects in the organ, and vancomycin and ceftriaxone are useful antibiotics that help treating a number of bacterial infections that may arise during the experiment. In addition, it is provided an infusion of the essential biomolecules: calcium gluconate for a mineral supplement and prostacyclin as a lipid contribution. A constant parenteral nutrition is also being mixed with the current perfusion liquid. This one contains the remaining essential biomolecules required by the liver. At the very beginning of the perfusion, an important parameter to be checked is the pH. Blood should be always within a range[5] and this cannot be beyond the limits. In order to have a starting proper pH value, an infusion of calcium bicarbonate is provided. This one lowers the pH value since this is the tendency showed by blood.

Replacement liquid

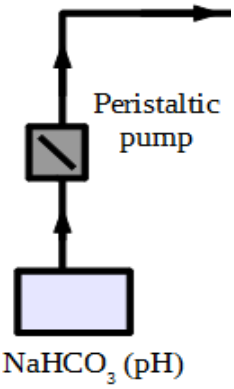


Depending on the organ's needs, the liquid for replenishment may vary from saline, sterile solution of sodium chloride in water, to a glycosilated solution. The latter refers to Viaflo Plasmalyte 148 (pH 7.4) from Baxter. It is commonly used as perfusion solution intravenously for patients. It provides high concentrations of glucose along with several electrolytes contribution.

As it has been discussed previously, variations of glucose appeared in the first perfusion experiences. This is the reason why Plasmalyte constitutes a good option to overcome a lack of glucose in the system.

Sodium bicarbonate

The pH control is based on the infusion of sodium bicarbonate (NaHCO3). Since the blood tends to lower its pH value, only this solution is needed because its presence in the system increases the pH depending on the volume added[6]. As it has been previously explained, the solution is directly connected to a peristaltic pump that allows its rapid addition every time the control of pH demands it. The solution used in our system is Sodium bicarbonate 1/6M Braun (B.Braun Medical SA, Barcelona, Spain).



[5] Blood pH range: 7.35 – 7.45.

[6] Ionic dissociation of the sodium bicarbonate: $NaHCO_3 \leftrightarrow 2Na^+ + HCO_3$.

4.2. Control software

A central PC-based program provides a user interface that allows the control of the whole system. This is developed within an integrated development environment (IDE) called GAMBAS 3. This provides a platform with all the facilities for software development. It is based on a dialect of the BASIC programming language. It constitutes a high-level programming languages whose design philosophy emphasizes ease of use. Gambas results a perfect option for the construction of the graphical user interface (GUI) used for the system. It allows the user to interact with electronic devices through graphical icons. This is possible since Gambas is object-oriented, where the objects are each of the devices that take part in the system. The components are able to send and to receive information.

Gambas provides a wide variety of options in programming designed to run on Linux and other Unix-like computer operating systems [14]. For the purpose of this project, Gambas is selected because of its design to build graphical applications specially using the Qt[7] toolkit, the one used here. This toolkit allows the creation of buttons, text boxes and a variety of components that provide additional features. These functionalities aid to create a user interface. The interface gives the user a quick idea about the current state of the system. The data is displayed and at the same time, it can be controlled or modified manually. However, the software behind is developed so that the whole circuit can adjust and behave autonomously.

The components of the circuit are communicated with the computer through a transparent wireless connection. This communication pathway is bidirectional. This is, current information about the circuit status goes to the PC and changes set by the user in the PC are sent as commands to the devices that compose it. The software for the control is fully developed aiming an easy-to-use interface for the user. It allows no interaction of the user with the circuit directly but only monitoring and commanding from the PC. The wireless connection is the work of the Xbee. This radio module integrates an Arduino board capable of gather all the information from each part of the system and send back to them commands in the opposite direction. It is connected to the computer through the standard USB communication protocol. At the same time, each of the devices are provided with a Xbee Wire Antenna module.

Gambas software

This is the primary program script that arranges the incoming data from all the devices that compose the system, and also communicates with each of them independently. The program carries out basic functions about data acquisition (data registration) and its further decoding for user understanding. Information comes through USB communication protocol provided by the main Xbee module that gathers all the data of the system wirelessly. The incoming data is received as a string. Each of the components provides a specific reference heading so that the main incoming string can be split. Once the identification is established, the codebreaking is done individually attending to the coding format. Each of the variables are stored and directly showed in the user

[7] Qt is a cross-platform application framework that is widely used for developing application software with a graphical user interface.

interface and also some of them are plotted in a graph as shown in Figure 9. This constitutes the graphic module of the control window. It provides the information for the user and allows him/her performing modifications and control mechanisms. These first functions that involve the information handling, are done for each of the devices that compose the circuit. The communication also involves the other direction pathway. This refers to the commands sent from the PC to the devices attending to the user's commands. These ones are printed back and sent through the same channel but in the opposite direction every time the system is adjusted or modified. This part that has been yet explained, constitutes the first part of the program.

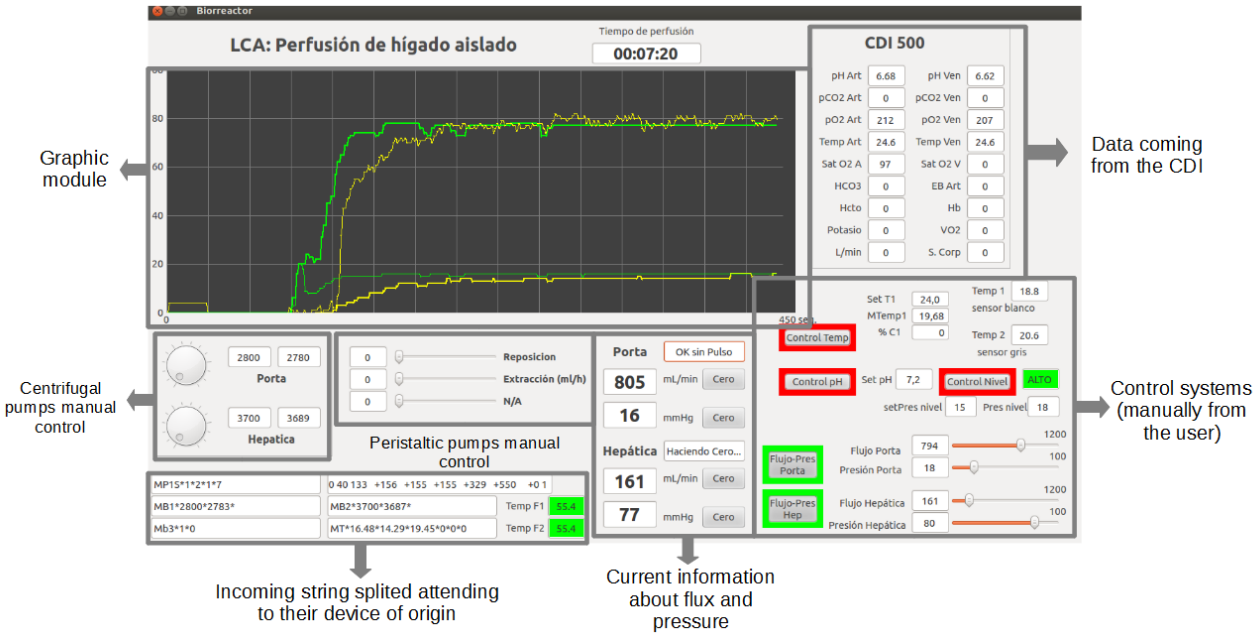


Figure 9. Window for user interface (each part specified)

A second part of the program is dedicated to the user manual modification. Velocity of the pumps for flow or pressure adjustment, temperature setting, and additional pumps activation for replenishment or filtration are provided with sliders, dials and scroll bars that assign the sending commands. These also need to be coded attending to the communication protocol in order to make possible proper understanding from each of the devices. Along with the remote control, the software provides an autonomic control platform once the conditions are specified by the user. The developed controls involve temperature, level, pH, flow and pressure. They are all activated at the pushed of a button. The click triggers the start of the control mechanism. Temperature control is based on the activation or deactivation of the thermostat based on the current temperature that is reported by the LM35 sensor. Level control works in a similar way, but it relies on both pressure and level capacitive sensors for the activation or deactivation of the peristaltic pump for replenishment. The pH control evaluates the difference between the actual pH value at the arterial path from the CDI with the set value introduced by the user. This is, once the current pH in the system is below the setpoint, the pH control starts. Its activation produces the activation of the peristaltic pump connected to this control system. This pump will start adding sodium bicarbonate to the system for a time that is proportional to the difference in pH values explained before. This control will stop only when the current arterial pH measured by the CDI equals or goes over the setpoint. The script part for the pH control software is detailed in the annexes. Finally, the last

control combines both flow and pressure. It constitutes the main part of the project that is being presented here.

The resistance that the liver presents, determines greatly the flow and the pressure it allows and needs to be fed with. This is the reason why the system needs to be able to adapt to any organ so to adjust each of the components to its characteristics. A proper blood intake assures the viability of the organ. It provides the right amount of blood needed to irrigate the whole organ and so on the cells it is composed of. It is required a control mechanism that works automatically once the liver is ready for perfusion. It should control the rate of the pumps for flow and pressure values depending on the resistance of the organ. And it needs to adjust continuously to any change that may occur during the operation.

Flow and pressure control: PID development

The flow-pressure control is also triggered at the pushed of a button. But in this case, there are two buttons for both the hepatic artery and the portal vein in such a way that the control systems can be activated independently. Previous control mechanisms rely on a simple activation/deactivation of a device, but this one uses a Proportional-Integral-Derivative (PID) loop feedback mechanism. A PID controller calculates the error value as the difference between the setpoint and the measured value. The first set chosen by the user will be the flow in each of the entries to the liver and this allows a margin for the pressure. Pressure values lie within the bounds of those specific for liver pigs[8]. The control elicits the activation of the centrifugal pumps at the specific velocity that meets the required flow and pressure desired setpoints. The feedback controls comes from the information received from pressure transducers and flowmeters measures.

As it has been previously explained, pressure and flow are important parameters that need to be accurately controlled during the experiment. Their current values depend on several other variables within the system, and they must keep always between their limiting values. Therefore, it is required a more complex control system than those discussed previously so to allow a more adjustable and precise control, which is the main purpose of the PID systems.

The flow-pressure control is able to control independently the flow going through the hepatic artery and the portal vein. This is, the velocity (rpm) of each centrifugal pump will be calculated as a function of the flow set value chosen by the user. This control of flow is limited by pressure values that the system cannot overpass when adjusting for the setpoints of flow. This is, the control of flow is the first one that is activated and depending on the changes in pressure that it produces, the control of pressure will start performing its action in order to correct this excess in pressure. Once this is corrected, the system will go back to the flow control. This is, inside the control of flow, there is a deeper control for pressure that assures that these two parameters (flow and pressure) are kept within the limits set by the user.

[8] Pressure at the portal vein: 12-17 mmHg.
Pressure at the hepatic artery: 75-80 mmHg.

This type of control requires an independent controller for each of the parameters. This is, a new evaluation of the coefficients of the controller and the dependency between them when the system needs to behave with characteristics of the organ that is being perfused.

A proportional-integral-derivative controller is a feedback mechanism able to adjust and produce a correction to any deviation happening in the system. For a correct control mechanism, the feedback mechanism needs at least three of the following components: a sensor, a controller and an actuator [15]. The sensors in this case are the flowmeters and the pressure transducers that determine the current state of the system. The actuator refers to the motor integrated in the centrifugal pumps that produces the direct adjustment of the system by modifying the rpm. Finally, the controller is in charge of generating the proper signal for the actuator to behave. This part corresponds to the control software developed in Gambas. It updates constantly the error value (difference) between the desired value and the current one. It produces the proper values needed for a correct adjustment with the operational computations characteristic of a PID controller.

The algorithm for the PID controller is given by three different parameters: the proportional, the derivative and the integral one. This three-term control sums up the action of each of the parameters in order to adjust the process. This adjustment follows the schematic shown in Figure 10. The parameters are computed independently and added up to get the new corrected value depending on the desired one (setpoint).

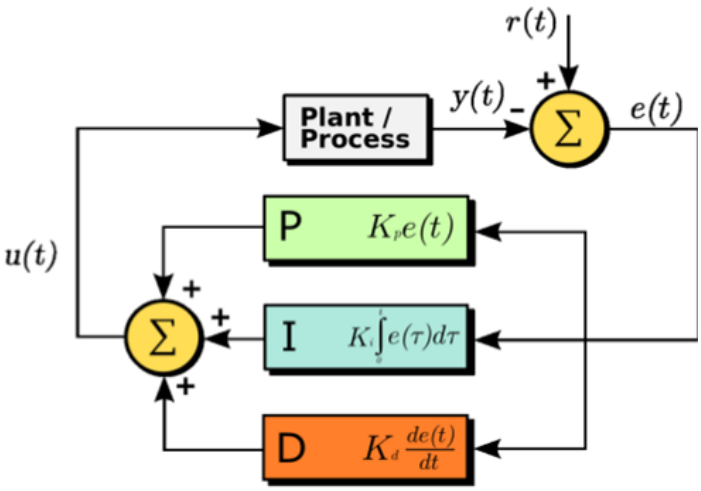


Figure 10. Block diagram of a PID controller in a feedback loop

The need of these three parameters is related to the complexity of the system. Each of them are computed attending to the correcting term they account for. The proportional one, is related to the actual error value and it is proportional to it. At the same time, this output value is multiplied by a coefficient value that is chosen by the user and fixed for the control (Fig.10). This Kp can produce an unstable system if it is chosen too high or in the other hand, it can produce a small control action that cannot overcome the existing error disturbances if it is chosen too low. The integral term has to do not only with the magnitude of the error, but with the duration of it. This is, the contribution of the integral term accelerates the movement of the process towards the setpoint. This second term has also a coefficient, Ki, that is chosen by the user and it is specific for the feedback control of the system. The integral part helps to correct the accumulated offset error that is produced by the

proportional term. Finally, the derivative term is intended to improve the settling time and the stability of the system. The action of the derivative term predicts the system behaviour but this action is seldom used in practical applications. This is the case in this project, in a first approximation, the flow-pressure control was thought to take into account these three parameters, this is, for a PID controller. But when working with the action of our control, this term did not enhance the feedback control. It was decided to set zero the Kd, the coefficient of the derivative term, keeping this term with no contribution to the final value for the velocity. This way, the system is less responsive and slower to reach the setpoint. The script part for this control is in more detail in the annexes. This is, the control of the flow and pressure is based now on a PI controller because the derivative of the error is not used.

Loop tuning

There exist different methods to adjust the coefficients of the control system. Designing and tuning a PID controller, or a PI in our case, may seem intuitive once it is well known how they affect the system. But it turns out to be a harder task specially when combining the action of two terms. Stability is one of the most important requirements. The response of the system should not oscillate around the setpoint. Since the response time in the system is short, about 2-3 minutes, the chosen method for the tuning was the manual one. This one does not require math calculations and it needs to be checked continuously by the program itself. The manual method can be accomplished as the user decides, but in this case, the procedure was the following. Since at the beginning the system was thought to be a PID controller, as it is discussed, it will be shown the modification from PID to PI. First of all, derivative and integral terms were set to zero and only the coefficient for the proportional term was being changed so to see the oscillation generated by the value chosen. This value is modified until the system shows the smallest oscillation possible. Once this coefficient is fixed, the Ki is adjusted so to overcome the offset. Once this is corrected, then the previous oscillation is improved and the system reaches sooner the setpoint. After these two terms, the derivative one should help the system to get to the setpoint faster than the previous approach but its contribution made an excessive response that destabilized the control system. This is the reason why finally it was decided to remove it and kept as zero. This procedure was done independently for the hepatic artery and the portal vein and for each of them, also independent analysis were done for pressure and flow.

The main equation that takes into account these three terms is being computed to get the new rpm calculated to adjust the system to the desired setpoint selected by the user (Fig.11). It is important to be aware that the user selects the desired values for flow and pressure, but these are changed upon adjustment of the velocity of the centrifugal pumps. The velocity exerted by them affects directly the flow and the pressure coming to the organ. This is, the velocity is changed based on the variations of these two parameters. The calculus for the equation shown in Figure 11 as much as the coefficients discussed, are presented in the script of the software program in the annexes.

$$\mathrm{V}(t) = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{d}{dt} e(t)$$

Figure 11. Final form for the PID algorithm

Once the corrected value for the velocity is computed, this is sent back to the pump through the wireless connection. This procedure is done at the same time for the hepatic artery and the portal vein. Each control can be activated independently but once both are working, the computations are being computed at the same time. The new velocity is printed in the serial port with the corresponding heading that specifies its destination (centrifugal pumps).

5. RESULTS

5.1. Water test experiment

The development of the software for the pressure-flow control is first checked with water. Water steps into the role of the perfusion liquid in the system for the test. No organ was used and instead, the endings of the conduits were implemented with some instruments that provided the corresponding resistances. These suited those characteristic resistances for the hepatic artery and portal vein of a liver.

The aim of the experiments rely on the maintenance of the system for long periods (24 hours). This is the reason why the tests with water were also performed so to achieve this objective. The following figures show the state of the water perfusion at three different time points.

Control stabilization

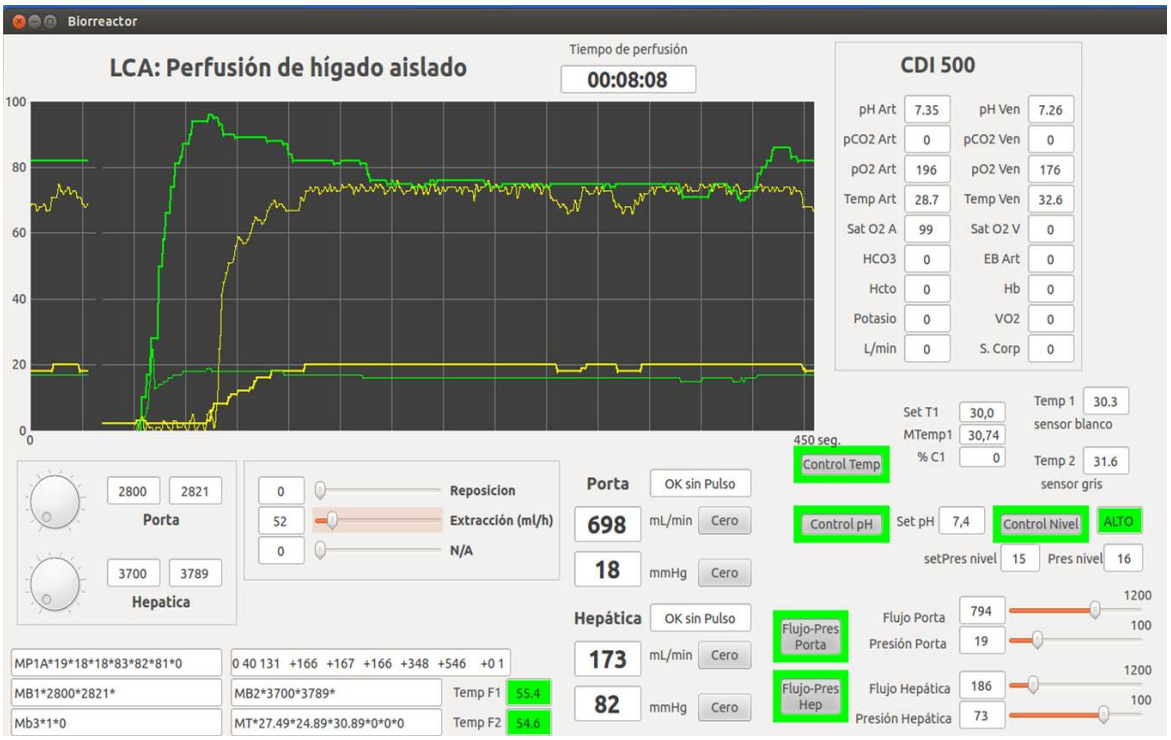


Figure 12. Screenshot of the interface at the beginning of the perfusion test using water (after 8 minutes)

Figure 12 exhibits in the graphic module how the system accomplishes the pressure-flow control. The curves in the graph are those corresponding to a PI control where the timing and oscillations have been optimized so to reach the set point the fastest and most accurate way. The system takes two minutes to reach the plateau and it presents very little or none oscillations. This is, there are almost no disturbances during this first part where the experiment is being stabilized.

Figure 13. Screenshot of the interface of the water test after 20 hours

Figure 14. Screenshot of the interface of the water test after 24 hours

Figures 13 and 14 include a real time visualization of the perfusion machine in the lab. This is done through a web camera connected to the computer. Since the perfusion was left active during night, a remote control was also installed in order to have access to the program from home at any time.

The setting conditions for pressure and flow introduced by the user are shown at the bottom right corner of the window. These do not change during the test and the current state of the system presents values around these ones. They are shown at the bottom central part of the window.

Parameter evolution

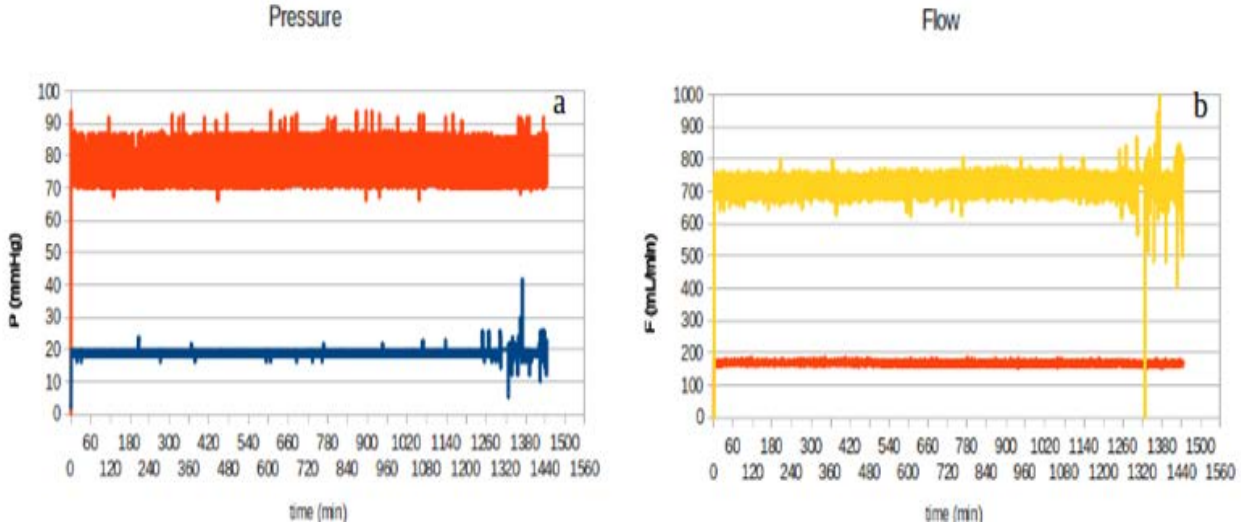


Figure 15. Graphs for the variations of pressure and flow in time during the test with water.
a) Hepatic artery pressure (blue) and portal vein pressure (orange), b) Hepatic artery flow (yellow) and portal vein flow (orange).

Figure 15 shows the stability of the two main parameters involved in the control. They remain within the set values chosen indicated in previous figures of the screenshots. The experiment lasted homogeneously constant for 24 hours as it is indicated in Figure 15 (1440 minutes).

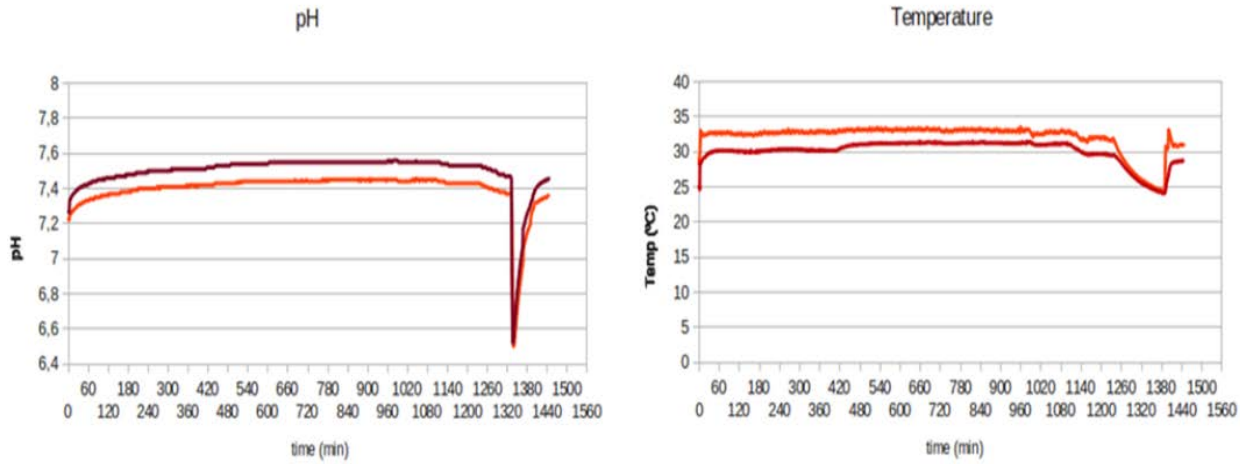


Figure 16. Graphs for the variations of temperature and pH during the test with water.

Other control mechanisms are also working at the same time for the complete working mode of the perfusion machine. Figure 16 presents the evolution in time of pH and temperature. These remain constant during the test. At minute 1320, the system was altered on purpose. A little quantity of HCl (0.5ml) was introduced in order to change the pH value (lowering it). This variation made the pH control activation start to recover the pH of the system within the set value chosen. The plot of the pH presents a rapid recovery for a large variation.

5.2. Perfusion experiments

Once the test with water showed successful results, the experiments were performed using a liver from a pig and blood from the same animal. Two perfusion experiments were carried out. For both of them, once the blood was already circulating in the circuit, the liver was introduced in the system. Figure 17 shows the real circuit with the liver placement for the experiment.

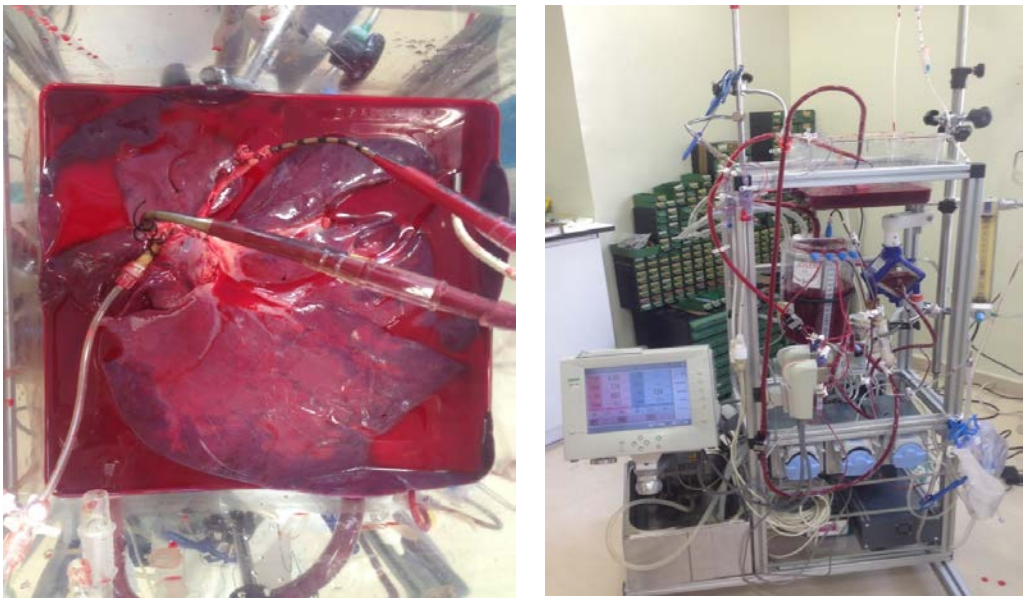


Figure 17. Pictures of the perfusion system and the liver in the lab during the experiment

Control stabilization

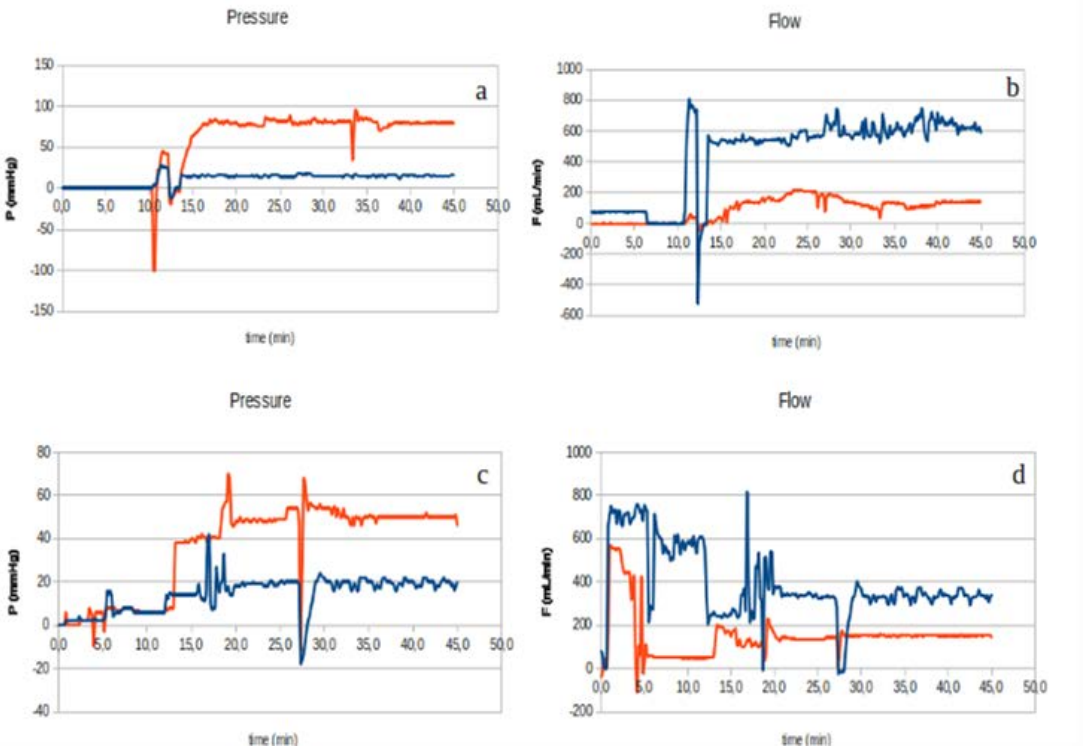


Figure 18. Graphs for the stabilization of pressure and flow at the beginning of each of the perfusions. a) Hepatic artery pressure (blue) and portal vein pressure (orange) for the first experiment, b) Hepatic artery flow (blue) and portal vein flow (orange) for the first experiment, c) Hepatic artery pressure (blue) and portal vein pressure (orange) for the second experiment, d) Hepatic artery flow (blue) and portal vein flow (orange) for the second experiment

The system takes some time until it stabilizes the pressure and flow control to the features of the liver. Figure 18 shows the characteristic curve of the PI control. At early time points, the values recorded correspond to the beginning of the experiment, when the blood is just circulating through the circuit but the organ has not yet been introduced. At minute 12 for both experiments, the liver is introduced and then the stabilization of the control starts. The values for both pressure and flow at each of the entries go growing up until they reach the set value chosen. Plots in Figure 18 present a rapid stabilization until they reach the plateau after little oscillations.

Figure 19. Screenshot of the interface during first perfusion (after 1 hour and 43 minutes)

In Figure 19 it is shown a screenshot of the first experiment. The current state of the portal pressure and hepatic flow reported in the user interface, match those introduced by the user. But when looking at the portal flow and the hepatic pressure chosen by the user, the current state values do not reach the setting ones. However they do fit its complementary value. This is, the portal vein is within the pressure set value, but this one do not allow the chosen ml/min due to its resistance. In this case, the control that is being working is the pressure one. Since the set value for the flow required a large value for the pressure, the second was the limiting one. The PI control works aiming to have the system within the set value for the pressure and this produces a reduction in the flow value for the portal vein. The opposite thing happens with the hepatic pressure. In this case, the activated control is the one for the flow. This achieves the setting value but it requires a lower value for the pressure that is not able to grow up to the user-chosen set value once the flow has been already reached. This is, the control does not enter into the pressure one.

Parameter evolution

Figure 19 shows the stabilization of the pressure-flow control that once it fits the characteristics of the organ more than an hour after the start of the experiment. This restriction is something that does not happen with water because it is already known the resistance and so on the flow and pressure needed since it has been previously modified attending to those values. Furthermore, in a real organ, the resistance changes because of the vasoconstriction and vasodilation of the vasculature of the liver. At the very beginning, right after the surgery, the liver

presents characteristics that go changing when the organ is being perfused again. Figure 20 shows the trend of both pressures and flows for the hepatic artery and portal vein. These two graphs show the changes that are being experienced by the system and how the control adapts to them autonomously keeping the organ alive maintaining its necessities. Furthermore, the variation experienced in the pressure values is due to a manual adjustment of the organ to the canuli for the entries. This enhanced a change in the characteristics of the liver.

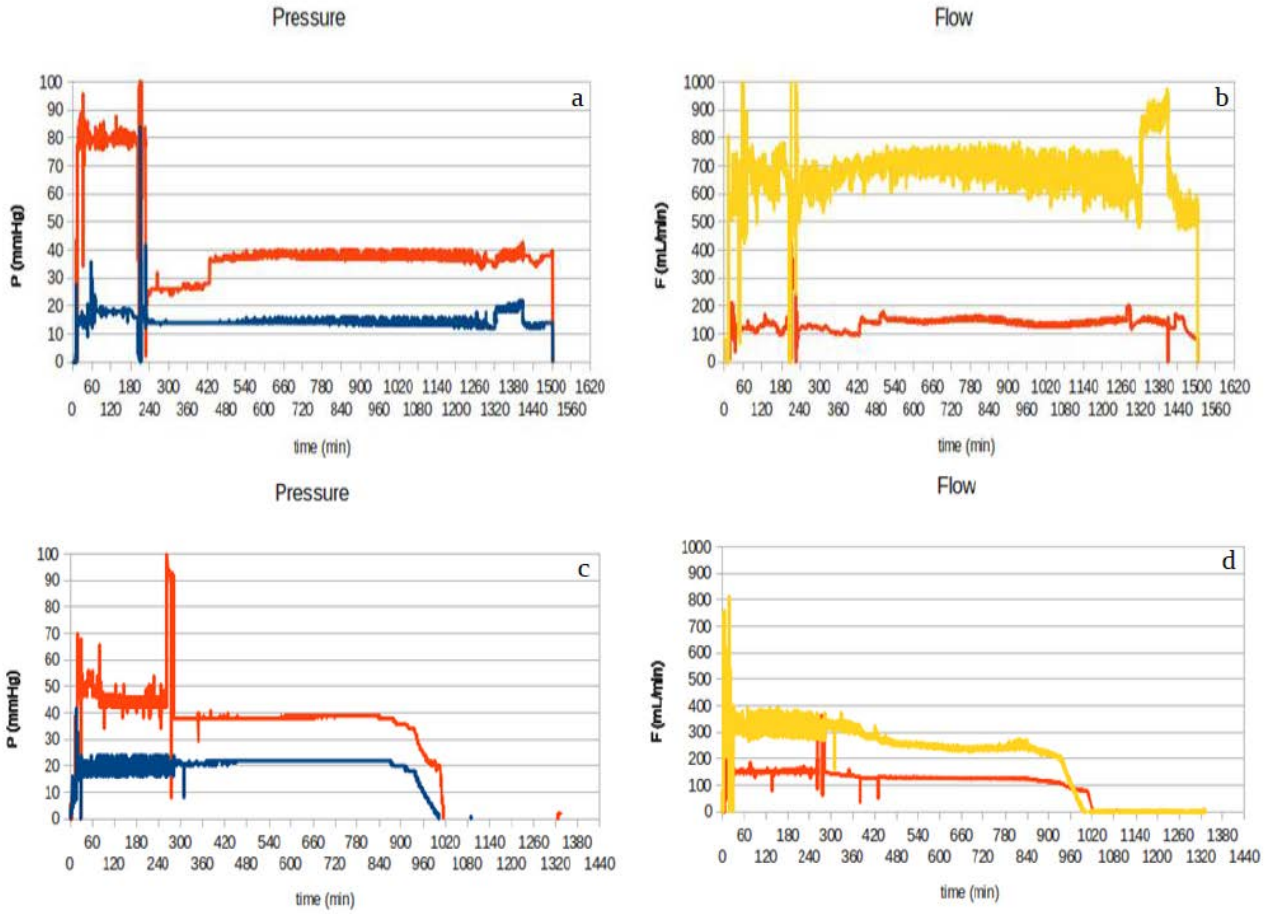


Figure 20. Graphs for the variations of pressure and flow along the second perfusion.
a) Hepatic artery pressure (blue) and portal vein pressure (orange) in the first experiment, b) Hepatic artery flow (yellow) and portal vein flow (orange) in the first experiment, c) Hepatic artery pressure (blue) and portal vein pressure (orange) in the second experiment, d) Hepatic artery flow (yellow) and portal vein flow (orange) in the second experiment

From Figure 20 it can be extracted that the first perfusion experiment lasted for 25 hours (1500 minutes). The x-axis indicates the time duration of the experiment in minutes. The second one, on the other hand, lasted for 14 hours (840 minutes) working at normal state. After that point, the system run out of replenishment liquid and then the experiment could not keep on with the total volume of the circuit and so the control couldn't maintain the conditions. In this case, the problem occurred during the night and then the system was not stopped until the next day although the pumps were deactivated from the remote control from home.

Gases evaluation

Along with the continuous registration of the monitored parameters during the experiments, the gases and other relevant components concentrations present in the blood are analyzed each hour. For each of the experiments, and attending to the different time durations of them, the following

figures show the evolution of these important parameters found in blood that give an idea of the function and viability of the organ.

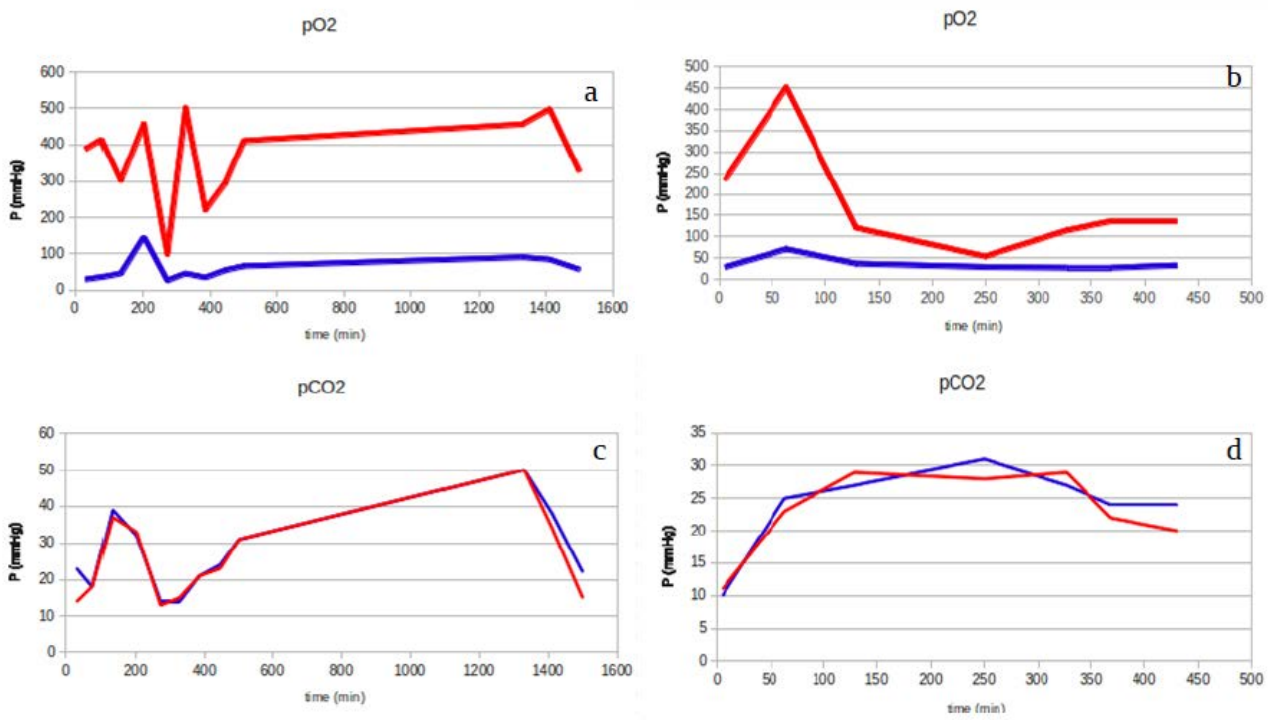


Figure 21. Graphs for the variations of the partial pressures of CO_2 and O_2 during first and second perfusion. Blue lines refer to the venous path and red to the arterial one. a) Partial pressure of O_2 along the first perfusion experiment, b) Partial pressure of O_2 along the second perfusion experiment, c) Partial pressure of CO_2 along the first perfusion experiment, d) Partial pressure of CO_2 along the second perfusion experiment

Figure 21 presents also an stabilization time at the beginning of the experiments. Once the system reaches the stable state, the values remain constant along the experiment until the end of the perfusion. Figures 21a and 21b show almost no change in the value of the pressure during the stable state. However, in the case of Figure 21a, the arterial pressure value of the O_2 presents a larger value than the characteristic one[9]. Since these gases are manually controlled by means of two rotameters, the value might be not accurately controlled. Figures 21c and 21d do not present values as stable as in the case of O_2, however, they range between those characteristic values for the partial pressure of CO_2 [10] after the stabilization period.

[9] Average partial pressure of O_2 in blood ranges between 80-100 mmHg.
[10] Average partial pressure of CO_2 in blood ranges between 35-45 mmHg.

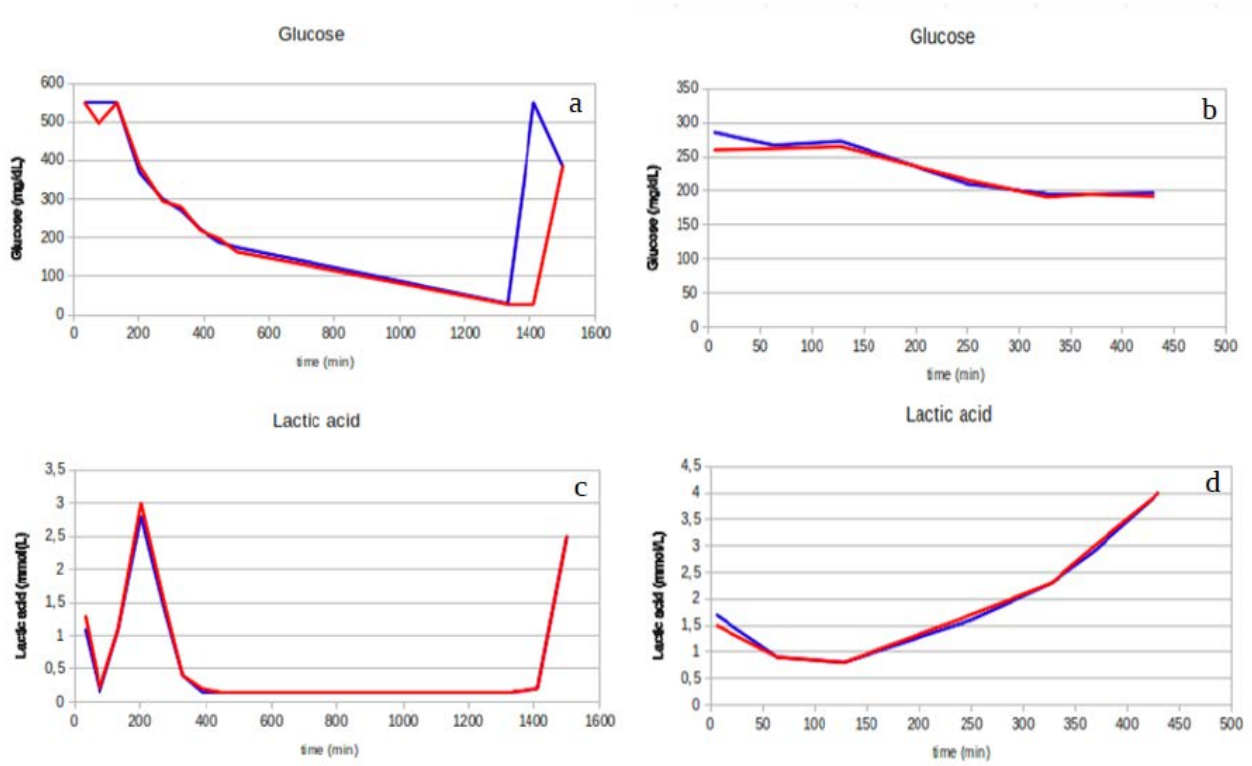


Figure 22. Graphs for the variations of the concentrations of glucose and lactic acid in blood during first and second perfusion. Blue lines refer to the venous path and red to the arterial one. a) Glucose concentration along the first perfusion experiment, b) Glucose concentration along the second perfusion experiment, c) Glucose concentration along the first perfusion experiment, d) Glucose concentration along the second perfusion experiment

Figure 22 show the relation between the glucose and the lactic acid concentration found in blood. Figures 22a and 22c belong to the first experiment. In this case, a continuous drop of glucose occurred over the perfusion, and these results in a decrease or no presence of acid lactate that is produced by glucose. During the perfusion, the replacement liquid used was not glycosilated and then the organ did not have the glucose contribution needed. In fact, this was realized near the end of the experiment. At minute 1400 the replacement liquid was changed for a glycosilated one and then these two parameters rose. Alternatively, in the second experiment, a glycosilated replacement liquid was used from the start. Figures 22b and 22d present stable values during the perfusion. A constant presence of glucose produced an increase concentration of lactic acid.

Biochemical analysis

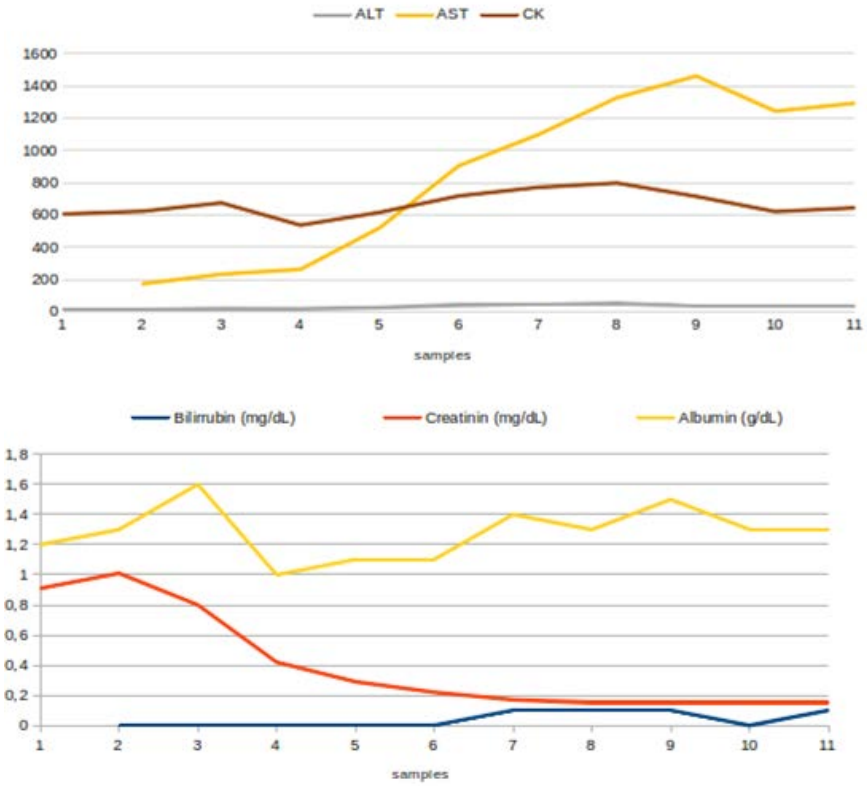


Figure 23. Graphs of the evolution of the biochemical parameters during the first perfusion. Enzymes in the upper plot: alanine transaminase (ALT), aspartate transaminase (AST) and creatine kinase (CK)

Figure 23 shows the evolution of three different enzymes. ALT is usually present in a high concentration in liver. Whenever the organ suffers a lesion, the concentration of ALT rises. This is, from the plot it is pointed out no lesions in the liver. Similarly, AST rises when there is a damage of the organ. At the end of the experiment, something may happened that altered this parameter. However, the stability of the remaining ones indicate good liver function. CK works the same as the previous enzymes. This is, its stability denotes no missfunction.

The bottom plot presents a low value of bilirrubine along the perfusion. This is a compound that appears when hemoglobin breaks down, this is, when hemolysis happens. Since bilirrubin is found in small quantities, this indicates no hemolysis within the circuit. Creatinine is usually a waste product generated by the muscles. Kidneys are the organs in charge of eliminating this compound from the body. In our case, the low value encountered during the experiment, indicates that the filtration system of the circuit (hemofilter) works well cleaning up these compounds. Finally, albumin is a compound present in blood and synthesized in the liver. It is homogeneously constant over the perfusion, which denotes that the liver keeps functioning. As previous parameters, this is encountered in lower values than the characteristic ranges for a common biochemical analysis for humans, but the experiment deals with porcine blood and the quantities keep stable and show liver functioning along the perfusion.

The biochemical analysis of the second perfusion did not show successful results since there was a problem at the end of the experiment. The circuit was stopped for some time and this caused the loss of meaningful measures in the blood.

6. DISCUSSION

The system that is being discussed here presents unique characteristics that have not been reported yet in any other similar research study to date. The state of the art about perfusion systems accomplishes more simple circuits (Figure 24). In general, these try to perfuse a liver in order to maintain or enhanced its function. But the circuits do not resemble that physiological function of the body that are related to the organ as ours does. In the following, it is analyzed some of the most recent publications that present similar studies to this project.

In the case of the studies of Diethard Monbaliu [17] and Kelvin G.M. Brockbank [16], they perfuse porcine livers hypothermically with UW solution[11]. This is a perfusion liquid specific to help preservation before organ transplantation. The pumps used for perfusing the liver are roller pumps with chosen values for pressure and flow lower (venous pressure around 8mmHg and arterial pressure around 25mmHg) than those specific for liver perfusion. They selected these conditions in order to avoid any damage to the endothelium due to shear stresses. However, these new values may cause loss of function in parts of the organ if the perfusion liquid is not able to reach the organ entirely. In contrast to our system, these roller pumps present an occlusive behavior since they exert a compression on the tube when moving the fluid inside. On the other hand, centrifugal pumps (the ones used in our circuits), move the fluid by means of a centrifugal force that creates a pressure difference with no occlusive behavior. This is, in case that the resistance at one of the entries of the organ rises, the roller pump keeps the fluid flowing despite of a complete tubal occlusion at the entry while centrifugal pumps stop the flow if the fluid is not able to move. This assures a safer behavior for the organ in case of occlusion problems during the perfusion. The pressure will increase in both cases but the difference relies on the hemolysis. Centrifugal pumps are commonly used in ECMO (Extracorporeal Membrane Oxygenation) circuits because they help avoiding hemolysis in sanguineous perfusion liquid circuits, while roller pumps usually present hemolysis depending on the degree of occlusion. However, the use of roller pumps in these two studies as well as in Bruinsma B.G.'s [5] is not a problem if and only if the perfusion liquid is UW solution. Finally, it is important to remark that when comparing the curves, both types of pumps exert a constant pressure and flow that is important for a control system as ours.

[11] University of Wisconsin solution.

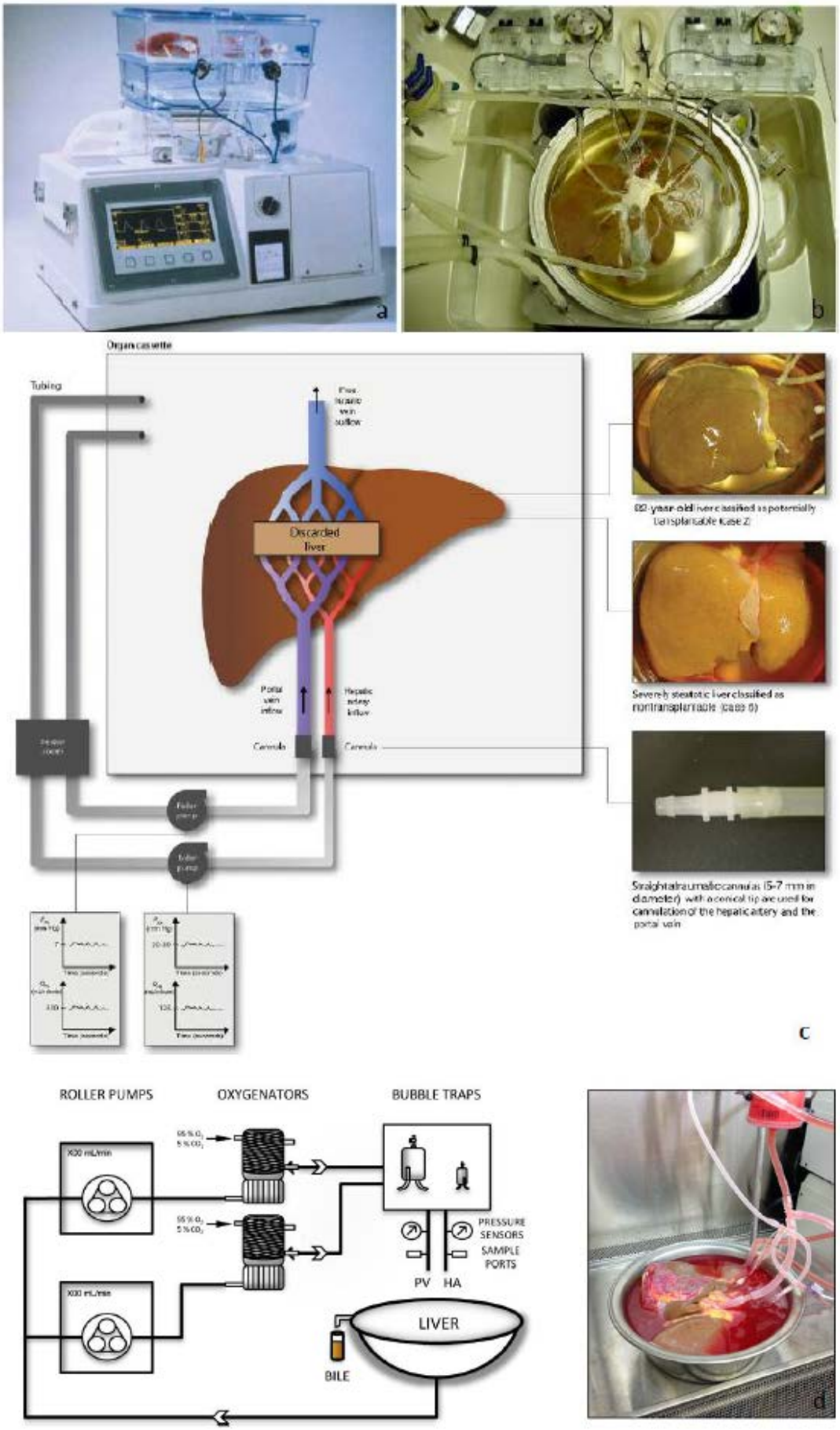


Figure 23. Perfusion circuits. a) Standard kidney perfusion machine used clinically [18], b) Prototype liver device during HMP [16], c) Schematic diagram of the HMP setup with separate roller pumps integrated into one device [17], d) Subnormothermic machine perfusion system [5]

On the other hand, other study of Shawn D St Peter about Extended preservation of non-heart-beating donor livers with normothermic machine perfusion [15] presents a much more complex perfusion circuit that looks alike to ours (Figure 24). This one uses a centrifugal pump although the perfusion liquid is again UW solution instead of blood. But this helps reaching characteristic values of pressure and flow at the entries of the organ. Arterial path is perfused with this type of pump whereas the venous one is fed passively from a reservoir by gravity, accomplishing a low pressure value. This is trying to develop a user-friendly circuit, and taking advantage of the gravity, the value for the venous pressure is kept constant easily, but it becomes very difficult to modify for control as our circuit does. Another centrifugal pump, as in our circuit, becomes an easy way to assess the pressure value for control. This is, modifying it depending on the circuit current state, and it still can be integrated in a parallel circuit with the arterial one. When comparing this system, it also shows the lack of the filtration system. This is the work of the kidneys in a real body. In our circuit, it helps clearing the blood and returns filtered blood that goes through the oxygenator. Since both are closed circuit systems, for long period perfusions (24h in both cases) should filter and clean up the perfusion liquid, even if it is UW solution what is being used since.

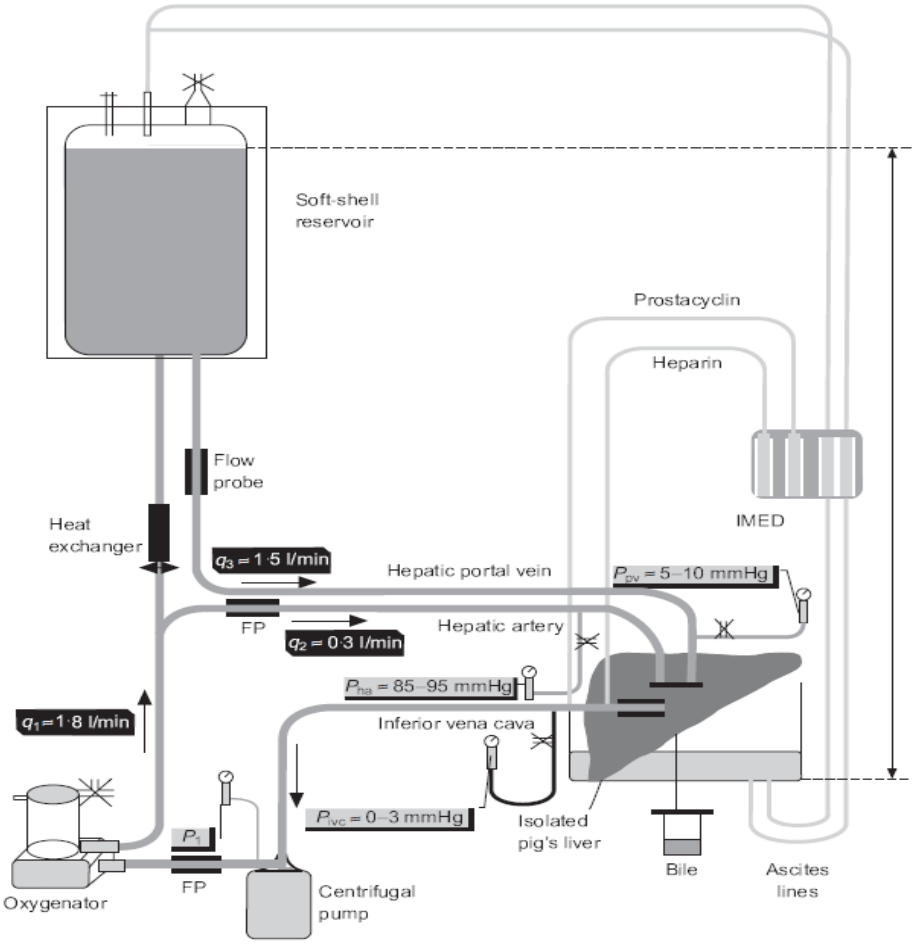


Figure 24. Extracorporeal perfusion circuit of Shawn D St Peter's study about *Extended preservation of non-heart-beating donor livers with normothermic machine perfusion* [15]

A main and most important difference that exists between our perfusion machine and those that have been already reported, lies on the integration of the PC control. This allows an autonomous adaptation of the system to the requirements and characteristics of the organ. Livers are not all the same and so the conditions vary depending on their features. In addition, the flow-

pressure control allows an autonomous adjustment that improves the manual setting shown in Bruinsma B.G.'s circuit (Figure 23b) [5]. Adjusting the peristaltic pumps for the maintenance of the pressure adds an extra workload to the user that can be avoided with the software control. Our method also allows for a constant measure of parameters. Arterial and venous blood gases and haemodynamics are recorded thanks to the CDI monitor, instead of a manual analysis every hour as in the case of Shawn D St Peter's study about Extended preservation of non-heart-beating donor livers with normothermic machine perfusion [15] for the assessment of function. Besides, the use of sensors as the flowmeters can record instantaneous current flow values without the need of calculations each hour as it happens in Diethard Monbaliu's research [17] or the resistances in Bruinsma B.G's [5].

In the perfusion experiments reported, the main objective differs from the one discussed here. The studies focus on a medical point of view in which from a warm ischemia state during surgery and the time the organ experiences it, the preserving period will vary so to obtain a viable organ for transplantation. The preserving period consist of different steps in each of the experiments. They may include a cold storage before perfusion or only a perfusion step hypothermic or normothermically. However, even though the aim varies, all of them have a perfusion step that could be improved. Our system presents an enhance option for control and organ adaptation that also has shown successful results in organ viability and function.

In fact, Kelvin G.M. Brockbank's research [16] expresses the thought that ex vivo normothermic perfusions for assessment of liver performance could be used as treatment to restore metabolic status of the organ, prior to transplantation. And Bruinsma B.G.'s study [5] considers the success of long normothermic perfusions over static cold preservation. This is what we have proved in our project, where although we begin from healthy organs (there is no need of metabolic restoration) the metabolic function is maintained during 24 hours.

Finally, results encountered in Shawn D St Peter's review on Liver and kidney preservation by perfusion [18] and Bruinsma B.G.'s [5] show that pressure-flow relations could reveal injury and predict organ function. These data of perfusion measurements could be defined so to distinguish a viable organ for transplantation from one destined to fail [18]. As they explained in the study, this parameter definition needs an increase in clinical experience with perfusion, but if they were perfectly defined, this could be implemented in our system. The software could include this organ viability evaluation. Once the perfusion is done and the parameters are recorded, the system could autonomously evaluate these values and tell the user whether the organ is valid or not for transplantation as a final step of the perfusion experiment.

7. CONCLUSIONS

- The software control implemented in the perfusion machine for the normothermic perfusion of an isolated liver, presents a control system able to maintain a constant and stable state.
- It shows the ability to homogeneously regulate the parameters that involve the correct performance of the experiment. Especially, those involving the pressure and flow experienced at the entries of the liver along with the total pH encountered in the circuit.
- The system provides a method that can adjust autonomously to the characteristics of the organ as well as to any changes occurring during the perfusion.
- It provides a user-friendly platform able to work on its own for long periods of time without the need of human surveillance.
- This design offers an improved method to actual systems that are currently used for perfusion experiments. And even presents an environment that allows the possibility of further implementations. This is, future software-controlled applications or measurements that could be helping the evaluation or clinical analysis of the organ within the perfusion research field.

8. BUDGET

The total budget of the project can be divided into different sections attending to the steps followed when reaching the final goal.

8.1. Acquisition budget

The construction of the perfusion machine requires the acquisition of each of the parts that compose it. Some of them are disposable devices and so a new one was needed for each of the perfusions (specified by ‘x2 perf’).

External part (visually exposed):

Oxygenator QUADROX-iD Pediatric …..... 3100 x 1 x 2perf = 6200€

Renaflo® II HF MinifilterTM Plus …..... 50 x 1 x 2perf = 100€

Propylene container '500-type cages+lid' (organ container) …….…..... 20 x 1 = 20€

VHK 2001 Venaous Hardshell Cardiotomy Reservoir ……..... 94 x 1 x 2perf =188€

Thermotronic II (thermostat) …..... 1240 x 1 = 1240€

Temperature sensors LM35 …..... 5.20 x 3 =15.60€

Emergency kit of ROTAFLOW system …..... 1450 x 2 = 2900€

VerderFLEX® EZ OEM pump …..... 268.83 x 3 = 806.49€

Alaris® CC Plus syringe pump …..... 625 x 2 = 1250€

Two Channel IBP OEM board EG 02000 …..... 250 x 1 = 250€

TrueWave Disposable Pressure Transducers …..... 20 x 3 x 2perf = 120€

OEM Ultrasonic Flow Measuring System DIGIFLOW-EXT1 ……...…. 1150 x 2 = 2300€

Ultrasonic Clamp-On Transducer …..... 1400 x 2 = 2800€

CDITM Blood Parameter Monitoring System 500 ..... 28156 x 1 = 28156€

CDI disposable sensors …..... 130 x 1 x 2perf = 130€

Conduits and scaffolding …..... 100 x 2perf= 200€

Total expenses on the external part: **46676.09€**

Internal part:

Electronic and mechanical material (Xbee, Arduino, electronic circuitry,

pumps development and construction, etc)..... 6000 x 1 = 6000€

Computational material for control (GAMBAS 3, PC, etc) …..... 1600 x 1 =1600€

Total expenses on the internal part: **7600€**

8.2. Performance budget

Once the perfusion machine was ready for its use, the number of experiments performed also required extra expenses. The line item budget is described as follows:

Perfusion experiments (2):

Pigs 'minipigs' …..... 320 x 2 = 640€

Operating room and animal cares …..... 3100 x 2 = 6200€

Biochemical analysis …..... 100 x 2 = 200€

Pathological anatomy technician …..... 50 x 2 = 100€

Replacement liquid and other infusions …..... 100 x 2 = 200€

Total expenses on perfusion experiments: **7340€**

Proffesional staff: 24.8€/h for an engineer

Research and project investigation ……...… 90h x 24.8€/h = 2232€

Software development: tests and optimization ……...…… 120h x 24.8€/h = 2976€

Perfusion experiments, results and data analysis …… 34h x 24.8€/h = 843.2€

Total expenses on professional staff: **6051.2€**

8.3. Total budget

The final budget of the project sums up the expenses on the external parts, internal parts, the perfusion experiments and the staff.

Total expenses on the external part …….…….. 46676.09€

Total expenses on the internal part ……. 7600€

Total expenses on perfusion experiments ……... 7340€

Total expenses on professional staff …… 6051.2€

Final budget of the entire project: 67667.29€

9. BIBLIOGRAPHY AND REFERENCES

[1] Le Gallois JJC. *Experiments on the principle of life*. Thomas M, Editor. Philadelphia: Conference Ed. 1813.

[2] Carrel A, Lindbergh CA. *The culture of whole organs*. Science. 1935; 81(2112): 621-3.

[3] Shumacker HB. *A surgeon to remember: notes about Vladimir Demikhov*. Ann Thorac Cardiovasc Surg. 1994; 58(4): 1196-8.

[4] Preservación de los órganos, almacenaje y transporte. http://donacion.organos.ua.es/submenu3/inf_sanitaria/proceso/preserva.asp [Accessed in July 2014]

[5] Bruinsma B.G., Yeh H., Ozer S., et al. Subnormothermic Machine Perfusion for Ex Vivo *Preservation and Recovery of the Human Liver for Transplantation.* American Journal of Transplantation. 2014; 14: 1400-1409

[6] Asencio JM, Vaquero J, Olmedilla L, García Sabrido JL. *"Small-for-flow" syndrome: shifting the "size" paradigm.* Med Hypotheses. 2013 May; 80(5): 573-7.

[7] Guyton and Hall. *Textbook of Medical Physiology.* Elsevier Inc, 12th Ed. 2012; 70: 837-841.

[8] Digestive System: Liver. http://www.innerbody.com/image_digeov/card10-new2.html#full-description [Accessed in July 2014]

[9] Michael H. Ross and Wojciech Pawlina. *Histología*. Médica Panamericana, 5th Ed. 2007; 624-640.

[10] VHK 2000/2001 Venous Hardshell Cardiotomy Reservoir http://www.maquet.com/us/product/VHK-2000-2001-Venous-Hardshell-Cardiotomy-Reservoir?parentNodeId=hlt5ls3x [Accessed in July 2014]

[11] Lespron Rfobles Mdel C, Molina Méndez J. *Terapia anticoagulante en la circulación extracorpórea / Anticoagulant therapy and cardiopulmonary bypass.* Arch Cardiol Mex. 2007; 77: S4:185-193.

[12] Alaris® CC Plus Syringe Pump with Guardrails® http://www.carefusion.co.nz/medical-products/inusion/alaris-system/alaris_cc_plus_syringe_pump_guardrails.aspx [Accessed in July 2014]

[13] CDI™ Blood Parameter Monitoring System 500 http://www.terumo-cvs.com/products/ProductDetail.aspx?groupId=1&familyID=47&country=1 [Accesses in July 2014]

[14] Gambas http://en.wikipedia.org/wiki/Gambas [Accessed in July 2014]

[15] S. D. St Peter, C. J. Imber, I. Lopez, D. Hughes and P. J. Friend . *Extended preservation of non-heart-beating donor livers with normothermic machine perfusion.* British Journal of Surgery 2002; 89: 609-616 .

[16] Kelvin G.M. Brockbank, Charles Y. Lee. *Ex vivo evaluation of porcine livers post-hypothermic machine perfusion at 4-6ºC and 12-14ºC* . Journal of Regenerative Medicine and Tissue Engineering 2012 .

[17] Diethard Monbaliu, Qiang Liu, Louis Libbrecht, Rita De Vos. *Preserving the Morphology and Evaluating the Quality of Liver Grafts by Hypothermic Machine Perfusion: A Proof-of-Concept Study Using Discarded Human Livers* . Liver transplantation 2012; 18:1495-1507.

[18] Shawn D St Peter, Charles J Imber, Peter J Friend. Review on l*iver and kidney preservation by perfusion* . The Lancet 2002; 359: 604-613.

10. ANNEXES

- Script part for the pH control:

```
 Public Sub TimerpH_Timer() 'Control de pH

 If TypeOf(Val(SetpH.Text)) = gb.Float Then ThpH = Val(SetpH.Text)
'Lee el valor asignado al set para compararlo con la medida actual de
pH
 If pHArt < ThpH Then
'El control comienza si el pH está por debajo del set impuesto
   dif = ThpH - pHArt '
   seg = dif / 0.03
'Segundos que la bomba estará activa en función de la distancia al pH
ideal y al volumen de carbónico expulsado por la bomba durante 1
segundo (experimental)
   seg = Round(seg, 0)
'Redondeo del valor de los segundos
   SetVelBV3 = 200
'Bomba en marcha a 200rpm
    Wait seg
'Tiempo que está activada la bomba
    SetVelBV3 = 0
'Para la bomba
    TimerpH.Delay = 120000 * seg
'La bomba estará parada un tiempo proporcional al tiempo que estuvo
encendida (estimado, approx)
  Endif
'El delay anterior (60000) que se tenía para el control, varía

End

Public Sub BControlpH_Click() 'Activa o desactiva el control de pH

  TimerpH.Delay = 60000
'El control de pH se realiza cada minuto
    If TimerpH.Enabled = False Then
'Una vez que se activa el control, el color del mismo se vuelve
verde, si no, rojo
    DAControlpH.Background = Color.Green
    TimerpH.Enabled = True
   Else
    DAControlpH.Background = Color.Red
    TimerpH.Enabled = False
  Endif
End
```

- Script part for the PID control of F-P:

```
Public Sub TimerControlHep_Timer()
''**Variables para el control de la arteria hepatica**
'control de la presion
  Dim i As Integer
  Dim h As Integer
  Dim ErrPrHPro As Integer
  Dim ErrPrHDif As Integer
  Dim ErrPrHInt As Integer
  Dim CoefHP As Float
  Dim CoefHD As Float
  Dim CoefHI As Float
  Dim MedPrH As Float

'control del flujo
  Dim ErrFHPro As Integer
  Dim ErrFHDif As Integer
  Dim ErrFHInt As Integer
  Dim CoefFHP As Float
  Dim CoefFHD As Float
  Dim CoefFHI As Float
  Dim MedF2H As Float

'******************Control Hepatica*****************************

CoefFHP = 0.05
CoefFHD = 0 '.0042 '.01 '.0001
CoefFHI = 0.0001 '.001 '.001

CoefHP = 0.02 '0.1
'Coeficientes PID
CoefHD = 0 '0.1
CoefHI = 0.001 '.003 '0.02

For i = 2 To 10
  ArrayPm2[i - 1] = ArrayPm2[i]
Next
ArrayPm2[10] = Pm2

For i = 1 To 10
MedPrH = MedPrH + ArrayPm2[i]
Next
MedPrH = MedPrH / 10
'Print "medH", MedPrH

For i = 2 To 10
ArrayF2[i - 1] = ArrayF2[i]
Next
ArrayF2[10] = Flujo2
```

```
 For i = 1 To 10
 MedF2H = MedF2H + ArrF2[i]
 Next
 MedF2H = MedF2H / 10
'Print "media", MedF1P

 ErrFHInt = MedF2H - SetControlF2
 ErrFHPro = Flujo2 - SetControlF2
 ErrFHDif = Flujo2 - F2Ant
 Print F2Ant

 ErrPrHPro = Pm2 - SetControlPm2
'Error proporcional
 ErrPrHDif = Pm2 - Pm2Ant
'Error diferencial
 ErrPrHInt = MedPrH - SetControlPm2
'Error integral
 Pm2Ant = Pm2

If MedPrH < SetControlPm2 Then
'* CONTROL PID DEL FLUJO DE LA ARTERIA HEPATICA (Flujo2);
SetControlF2 es el flujo que queremos en la Hepatica *
  VelControlHep = VelControlHep - (CoefFHP * ErrFHPro) - (CoefFHD *
ErrFHDif) - (CoefFHI * ErrFHInt)
  Print VelControlHep, CoefFHP * ErrFHPro, CoefFHD * ErrFHDif,
CoefFHI * ErrFHInt
  Debug "Control flujo hepática"
Else                                                    '* CONTROL
PID DE LA PRESION DE ARTERIA HEPATICA (Pm2); SetControlPm2 es la
presion que queremos en la Hepatica *
  VelControlHep = VelControlHep - (CoefHP * ErrPrHPro) - (CoefHD *
ErrPrHDif) - (CoefHI * ErrPrHInt)          'Control de la velocidad
  Debug "Control presión hepática"
Endif

If VelControlHep < 0 Then VelControlHep = 0
SetVelB2 = VelControlHep
If SetVelB2 > 50 Then SetVelB2 = 50
'Límite de velocidad máxima: 5000rpm
If SetVelB2 < 0 Then SetVelB2 = 0
'Límite de velocidad mínima: 0rpm

'**************** TERMINA Control Hepatica**********************

End
```

```
Public Sub TimerControlPorta_Timer()

''**Variables para el control de la vena porta**
'control de la presion
  Dim i As Integer
  Dim h As Integer
  Dim ErrPrPPro As Integer
  Dim ErrPrPDif As Integer
  Dim ErrPrPInt As Integer
  Dim CoefPP As Float
  Dim CoefPD As Float
  Dim CoefPI As Float
  Dim MedPrP As Float
'control del flujo
  Dim ErrFPPro As Integer
  Dim ErrFPDif As Integer
  Dim ErrFPInt As Integer
  Dim CoefFPP As Float
  Dim CoefFPD As Float
  Dim CoefFPI As Float
  Dim MedF1P As Float

''**** CONTROL POR FLUJO DENTRO DE UNOS LÍMITES DE PRESIÓN *******
'Se controlan independientemente el flujo de la arteria y de la porta. Es decir, la velocidad de las bombas centrífugas estará en función del set del flujo que se haya establecido.
'Pero el control por flujo dejará de estar activo cuando la presión supere los límites establecidos como set de presion. En ese momento se activará el control de presión.
'Éste se encargará de que la presión no sobrepase estos límites al variar las vueltas de las bombas. Los flujos cambiarán y ya no serán los establecidos en el set previamente.

' ********************** Control Porta*******************
CoefFPP = 0.002
CoefFPD = 0 '0.0001
CoefFPI = 0.0001

CoefPP = 0.03 '0.1
'Coeficientes PID
CoefPD = 0 '0.1
CoefPI = 0.01 '0.02

For i = 2 To 10
ArrayPm1[i - 1] = ArrayPm1[i]
'Print "arrP", ArrayPm1[i]

Next
ArrayPm1[10] = Pm1
```

```
For i = 1 To 10
  MedPrP = MedPrP + ArrayPm1[i]
Next
MedPrP = MedPrP / 10
 ' Print "med", MedPrP

For i = 2 To 10
  ArrayF1[i - 1] = ArrayF1[i]
Next
ArrayF1[10] = Flujo1

For i = 1 To 10
  MedF1P = MedF1P + ArrF1[i]
Next
MedF1P = MedF1P / 10
'Print "media", MedF1P

 ErrFPPro = Flujo1 - SetControlF1
 ErrFPDif = Flujo1 - F1Ant
 ErrFPInt = MedF1P - SetControlF1

 ErrPrPPro = Pm1 - SetControlPm1
'Error proporcional
 ErrPrPDif = Pm1 - Pm1Ant
'Error diferencial
 ErrPrPInt = MedPrP - SetControlPm1
'Error integral
 Pm1Ant = Pm1

If MedPrP < SetControlPm1 Then       '* CONTROL PID DEL FLUJO DE LA
VENA PORTA (Flujo1); SetControlF1 es el flujo que queremos en la
porta *
 VelControlPorta = VelControlPorta - (CoefFPP * ErrFPPro) - (CoefFPD
* ErrFPDif) - (CoefFPI * ErrFPInt)
' Print VelControlPorta, CoefFPP * ErrFPPro, CoefFPD * ErrFPDif,
CoefFPI * ErrFPInt
 Debug "Control flujo porta"
Else                                  '* CONTROL PID DE LA PRESION DE
VENA PORTA (Pm1); SetControlPm1 es la presion que queremos en la
Porta *
  VelControlPorta = VelControlPorta - (CoefPP * ErrPrPPro) - (CoefPD
* ErrPrPDif) - (CoefPI * ErrPrPInt)      'Control de la velocidad
  Debug "Control presión porta"
Endif

If VelControlPorta < 0 Then VelControlPorta = 0
SetVelB1 = VelControlPorta
If SetVelB1 > 50 Then SetVelB1 = 50
'Límite de velocidad máxima: 5000rpm
If SetVelB1 < 0 Then SetVelB1 = 0
'Límite de velocidad mínima: 0rpm
'************* TERMINA Control Porta ****************************

End
```