

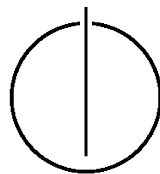
FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Doktors der Naturwissenschaften (Dr. rer. nat.)

**Automation of a Portable Extracorporeal
Circulatory Support System with the Use of
Adaptive Fuzzy Controllers**

Alejandro Mendoza García



TECHNISCHEN UNIVERSITÄT MÜNCHEN
Lehrstuhl für Echtzeitsysteme und Robotik

Automation of a Portable Extracorporeal Circulatory Support System with the Use of Adaptive Fuzzy Controllers

Alejandro Mendoza García

Vollständiger Abdruck der von der Fakultät der Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Nassir Navab
Prüfer der Dissertation: 1. Univ.-Prof. Dr. Ing. habil. Alois Knoll
2. apl.Prof. Dr. Med. Robert Bauernschmitt

Die Dissertation wurde am 04.10.2012 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 23.04.2013 angenommen.

Zusammenfassung

Die frühe Behandlung mit Extrakorporalen Kreislauf-Unterstützungssystemen bei Patienten mit kardiogenem Schock wirkt sich positiv auf den weiteren Verlauf aus und kann einem Multi-Organversagen entgegenwirken. Um eine frühe Behandlung zu ermöglichen, ist eine kontinuierliche Überwachung des Patienten durch geschulte Kardiotechniker notwendig. Unter Notfallbedingungen kann eine ungeteilte Aufmerksamkeit für den Patienten nicht garantiert werden womit es zu Behandlungsfehlern kommen kann.

Durch die Automatisierung einer portablen Herz-Lungen-Maschine kann eine optimale Durchblutung mit minimalem Arbeitsaufwand für das Personal erreicht und ein sicherer Transport zum Krankenhaus ermöglicht werden. Der Schwerpunkt dieser Arbeit ist die Entwicklung eines adaptiven und robusten Regelungssystems, das anhand der momentanen Perfusion des Patienten die Herz-Lungen-Maschine steuert. Das resultierende System muss hochdynamisch sein, um sich an verschiedene Situationen und Patienten anpassen zu können und zu jeder Zeit die Sicherheit des Patienten garantieren.

Um ein solches Regelungssystem zu entwickeln wurde zuerst ein Tiermodell verwendet, um die notwendigen Signale bei der extrakorporalen Zirkulation zu analysieren. Im nächsten Schritt wurden diese Signale verwendet, um ein mathematisches Modell zu entwickeln. Dieses mathematische Modell beinhaltet ein Herz-Kreislaufsystem mit angeschlossener Herz-Lungen-Maschine, ein Gas-Austausch-Modell und ein Medikamentenmodell. Dieses Modell wurde in eine Simulationsumgebung integriert, mit der das Regelungssystem evaluiert und verbessert werden kann.

Das Regelungssystem basiert auf Fuzzy Logik. Diese ermöglicht, das Wissen von erfahrenen Kardiotechnikern leicht in Regelvorschriften darzustellen. Da die Vorbedingungen und Reaktionen der Patienten von Fall zu Fall anders sind ist ein adaptiver Mechanismus notwendig, um den initialen Fuzzycontroller an die Bedürfnisse des jeweiligen Patienten anzupassen.

Ein Software-Framework wurde entwickelt, das die schnelle Implementierung eines Regelungssystems ermöglicht. Dieses Framework kann nicht nur für die Automatisierung der Herz-Lungen-Maschine verwendet werden, sondern kann als Basis für verschiedene Regelungssysteme in der Medizin mit ähnlichen Anforderungen dienen. Verschiedene Simulationen werden dargestellt, um die Leistungsfähigkeit des Regelungssystems und des adaptiven Mechanismus zu zeigen. Zusätzliche Simulationen zeigen das Verhalten des HLM Regelungssystems in verschiedenen Patientenszenarien.

Abstract

Patients suffering from cardiogenic shock may benefit with an early application of a portable Extracorporeal Circulatory Support System (ECSS) preventing multi organ failure. This however requires the presence and constant supervision of the patient by trained personal at the emergency site. Under these circumstances full attention to the patient may not be guaranteed and operation errors may occur.

With the automation of the portable ECSS optimal perfusion may be achieved with minimal workload for the human operator allowing the safe transportation of the patient to the hospital.

The focus of this thesis is the development of an adaptive and robust control system that regulates perfusion based on online data of the patient. While the system needs to be highly dynamic, so that it is able to adapt to different situations, it must ensure maximal patient safety at all times.

To develop such control system first an animal model was used to analyze the type of signals acquired during extracorporeal circulation. This information was used as a reference for the creation of a mathematical model. The model includes a cardiovascular system undergoing extracorporeal circulation, a gas exchange model and a medication model. This was integrated into a simulation system that could be used for the creation and evaluation of the designed controller.

Fuzzy logic was considered as a control mechanism allowing the easy creation of rules based on the knowledge of trained perfusionists. Since patient pre-conditions and reactions will be different from one case to another an adaptive mechanism is proposed to modify the existing controller and adapt to the specific needs of the patient.

A software framework was developed allowing a fast implementation of the control system. This framework was created not only focusing on the automation of the ECSS but also to serve as a basis for the development of control systems for other medical devices with similar requirements.

Several simulations are presented showing the performance of the fuzzy controller with the proposed adaptive mechanism. Additional simulations show the response of the designed ECSS controller under different patient scenarios.

Acknowledgments

I want to give thanks to my advisor Prof. Dr. Alois Knoll for giving me the opportunity of working in this thesis, for his support and confidence. I would also like to thank Prof. Dr. Robert Bauernschmitt for taking the time to review my work and acting as a second advisor. From the department of Robotics and Embedded Systems I would also like to thank Dr. Schrott for his constant support and advise, all the colleagues from the department for becoming good friends and sharing great moments. Furthermore I also want to thank the people of the German Heart Center for giving a good working environment. Special thanks to Dr. Ulrich Schreiber, Dr. Stefan Eichorn, Bennedikt Baumgartner, Nicole Sprunk, Annemarie Stroh, and the students that worked for a short period in the Heart Center.

I would like to thank my parents Ofelia and Ildefonso, who have given me high values, and have allowed me to come this far. For their full support and trust in my decisions. To my sister, and family from whom it has been difficult to be far from. My friends from Munich, Mexico and all over the world, who even at the distance continuously helped and encouraged me giving also great moments in between to relax and enjoy life making this an amazing experience.

I dedicate this thesis to my grandfather, who also encouraged me when being close and in the distance, he showed strength even in the last moment.

Contents

Zusammenfassung	v
Abstract	vii
I. Introduction	1
1. Motivation	1
2. Objective	2
3. Methodology	3
4. Thesis Overview	4
II. Physiological and Medical Background	7
1. Cardiovascular System	7
1.1. Heart	7
1.2. Systemic Circulation	8
1.3. Cardiac Cycle	9
1.4. Gas Exchange	10
1.5. Heart and Circulatory System Failures	11
2. Extracorporeal Circulation	13
2.1. History	13
2.2. ECC for Cardiac Resuscitation and Life Support	16
2.3. Components of Extracorporeal Circulation	16
3. Perfusion	20
3.1. Blood Flow	20
3.2. Arterial Pressure	21
3.3. Hemodilution and Hematocrit	21
3.4. Gas Exchange	22
3.5. Temperature	24
3.6. Anesthesia and Medication	25
4. Biomedical Signals and Patient Parameters	26
4.1. Blood Pressure	26
4.2. Blood Flow	27
4.3. Heart Beat and Electrocardiogram	27
4.4. Gas Exchange	29

4.5.	Temperature	31
4.6.	Body Surface Area	31
4.7.	Systemic Vascular Resistance	32
4.8.	Summary of Measured Signals and Normal Values	32
III. Experimental and Mathematical Models		33
1.	Experimental Model	33
1.1.	Experimental Procedure	33
1.2.	Experiment Protocol	35
1.3.	Data Acquisition	37
1.4.	Experimental Results	38
1.5.	Remarks	51
2.	Mathematical Model of Extracorporeal Circulation	52
2.1.	Previous Cardiovascular and ECC Mathematical Models	52
2.2.	Mathematical Model Overview	52
2.3.	Cardiovascular Model	52
2.4.	Gas Exchange Model	55
2.5.	Extracorporeal Circulatory Support System	57
2.6.	Blood Concentration and Medication Model	61
2.7.	Patient Disorders and Possible Complications	62
3.	Simulation	64
3.1.	Simulation Environment	64
3.2.	Simulation Results	68
3.3.	Conclusion	73
IV. Automation of the Extracorporeal Circulatory Support System		75
1.	Fuzzy Control	75
1.1.	Fuzzy Sets	76
1.2.	Fuzzy Control Structure	77
1.3.	Mamdani Controller	78
1.4.	PI and PD Fuzzy Controller	80
2.	Multiple Input Adaptive Fuzzy Controller	82
2.1.	Model Reference Control	82
2.2.	Knowledge Controllers	83
2.3.	Adaptive Controller	83
2.4.	Learning Mechanism	84
2.5.	Adapting Mechanism	85
2.6.	Performance evaluation	86
2.7.	Fuzzy Supervisor and Diagnosis	86
3.	System Implementation	88
3.1.	System Description	88
3.2.	System Requirements	90
3.3.	Framework Implementation	91
3.4.	Component Based Development	91
3.5.	Embedded System Implementation	98

4.	Portable ECSS Control	102
4.1.	Parameters	102
4.2.	Centrifugal Pump Control	103
4.3.	Gas Blender	105
4.4.	Medication	107
4.5.	Implementation	108
V. Control and Simulation Results		109
1.	Control Learning and Adaptation	109
1.1.	Adapting without previous knowledge	110
1.2.	Learning without adaptation	110
1.3.	Learning and adaptation test	111
1.4.	Control mechanism comparison	111
2.	ECSS Control Results	112
2.1.	Case of beating heart and high pressure	112
2.2.	Case of non beating heart and low pressure	116
2.3.	Control Comparison	119
VI. Discussion and Conclusions		121
1.	Discussion	121
2.	Conclusion	125
3.	Future Work	126
Bibliography		129
Appendix		140
A. Detailed Descriptions		141
B. Developed Software		144
C. Mathematical Model (MML)		153
D. ECSS control		171

List of Figures

I-1.	Thesis Overview	4
II-1.	Heart anatomy [1].	8
II-2.	Cardiovascular System	9
II-3.	Gibbon creator of the first Heart-Lung Machine	14
II-4.	Model II Manufactured by IBM	14
II-5.	Different machines for extracorporeal circulation.	15
II-6.	Connection of ECSS with patient	16
II-7.	Extracorporeal Support System components	17
II-8.	Types of blood Pumps	17
II-9.	Relation of Hemoglobin Saturation and Oxygen Partial Pressure	23
II-10.	Strain gauge blood pressure transducer.	27
II-11.	Blood flow transducer	28
II-12.	QRS Complex.	28
II-13.	Pressure Volume and ECG curves	29
II-14.	Pulse oximetry principle [2]	30
II-15.	Blood gas analyzers	31
III-1.	Experimental setup	34
III-2.	Experimental protocol	35
III-3.	Arterial and venous cannula connected to femoral artery and vein.	36
III-4.	Pressure sensor positioning	37
III-5.	AutoMedic Platform.	38
III-6.	Steady states during experiment.	39
III-7.	Experiment analysis of heart rate, pressure and flow before ECC, during ECC with beating heart and without beating heart.	41
III-8.	Results pressure sensor positioning.	42
III-9.	Pressure sensors	44
III-10.	Change of centrifugal pump speeds with active heart in different experiments.	44
III-11.	Change of centrifugal pump speed in experiment 3 at different stages of the protocol.	45

III-12.	Oxygen and Carbon Dioxide partial pressures in relation to MAP and EFR	46
III-13.	Response to vasoconstrictor for different experiments.	47
III-14.	Response to vasoconstrictor in single experiment.	48
III-15.	Vasodilator	49
III-16.	Obstruction arterial cannula. (Left = Closing / Right = Opening)	50
III-17.	Obstruction venous cannula.(Left = Closing / Right = Opening)	51
III-18.	Mathematical model. Subsystem diagram.	53
III-19.	Complete Cardiovascular Model	54
III-20.	Six compartment model used for gas exchange and medication model.	56
III-21.	Experimental Setup	57
III-22.	ECSS components	58
III-23.	Pump Head Calculation	60
III-24.	Pump static and dynamic constant approximation	60
III-25.	Complete Model	61
III-26.	Viscosity vs Hcrit vs Temperature	63
III-27.	Patient Simulation under ECC	64
III-28.	JSim	65
III-29.	Simulation environment user interface	66
III-30.	Simulation Frames	67
III-31.	Simulation interconnection	67
III-32.	Arterial Pressure	68
III-33.	ECSS at different RPM.	69
III-34.	Medication concentration distribution in different compartments.	70
III-35.	Vasoconstrictor effect in simulation model and experiment.	71
III-36.	Vasodilator effect in simulation model and experiment.	72
III-37.	Component resistance comparison.	73
IV-1.	Membership Function	76
IV-2.	Overlap configuration in fuzzy sets	77
IV-3.	Focus vs regular distribution	77
IV-4.	Basic Fuzzy Controller	78
IV-5.	Rule evaluation with min max operators	79
IV-6.	Fuzzy PI and PD Control 3D Surface (normalized inputs [-1 1])	81
IV-7.	Adaptive control architecture.	82
IV-8.	Reference signal.	83
IV-9.	Learning mechanism from knowledge controllers.	84
IV-10.	Adaptive Mechanism.	85
IV-11.	System Components	88
IV-12.	Base Object.	91
IV-13.	Sum object: basic parameters	93
IV-14.	Sum object with view parameters	93
IV-15.	Graphical representation of Sum object.	93
IV-16.	Object Libraries	94
IV-17.	Device specific object library	94
IV-18.	App.xml	95
IV-19.	Objects1.xml	96

IV-20.	Objects2.xml	96
IV-21.	Calculate tree of program	97
IV-22.	TS-7300 Board	99
IV-23.	TS-7300 Board Enclosure	99
IV-24.	Software Layer	100
IV-25.	Supervisor system	100
IV-26.	Main window of ECSS control system	102
IV-27.	Four types of error with respect to reference model.	104
IV-28.	Pump Control with Knowledge Base controllers and Adaptive Controller .	105
IV-29.	Blender FiO2 Control	106
IV-30.	Blender Gas Flow Control	107
IV-31.	Medication Control	108
V-1.	Results of control mechanisms	110
V-2.	Results of learning and adaptive controller.	111
V-3.	Comparison between control mechanisms.	112
V-4.	Hemodynamic results	114
V-5.	Gas exchange results	115
V-6.	Hemodynamic results	117
V-7.	Gas results	118
V-8.	Comparison between control mechanisms for first case of high pressure and beating heart.	119
V-9.	Comparison between control mechanisms for second case of low pressure and heart stopped.	119
VI-1.	Future applications	126
B-1.	Object Libraries	144
B-2.	AMPacket Structure.	145
B-3.	Dynamic Channels.	146
B-4.	Scope	147
B-5.	Fuzzy Control Viewer	148
B-6.	Daq device drivers	150
B-7.	Serial device drivers	150
B-8.	Simulation Manager and Simulation Engine	151
B-9.	Frame contents	152

List of Tables

II.1. Measurement signals with normal ranges.	32
III.1. Component Constants	59
IV.1. Representation of inputs and outputs	77
IV.2. Rule base of a Fuzzy-PI	80
IV.3. Rule base of a Fuzzy-PD	80
IV.4. Control Parameters	103
V.1. Number of rules created for each controller after learning and adaptation. .	120
B.1. Implemented Objects, derived from BaseObject	147
B.2. Component Interfaces	149

Introduction

Contents

1. Motivation	1
2. Objective	2
3. Methodology	3
4. Thesis Overview	4

1. Motivation

Cardiovascular disease is a major causes of death in the world, being ischemic heart disease one of the main causes of mortality[3, 4]. In Germany, approximately 300,000 people die annually as a result of coronary heart disease. 10% of these deaths result from cardiogenic shock. Normal procedure for patients suffering from cardiac arrest is the application of cardiopulmonary resuscitation (CPR), however for some cases this treatment may result ineffective and produce further complications such as recurrent cardiac arrest due to low cardiac output along with haemodynamic instability, as a consequence producing multiple organ dysfunctions and increased risk of death.

In the cases such as coronary artery blockage or valve dysfunction the only possible treatment is through surgical intervention in the operation theater. However the time needed for transportation to the hospital might be too long for the patient to be kept alive without having proper blood circulation. For these cases the use of a system capable of generating extracorporeal circulation may help in saving these patients by restoring blood circulation and providing proper organ perfusion. An oxygenator may introduce oxygen into the blood and remove the excess of carbon dioxide. This system could either work together with the beating heart, reducing its load and therewith decrease the risk of further heart attack or completely replace the function of the heart in the cases of complete cardiac arrest, generating full body blood circulation.

Today in the hospital cardiac surgery involving extra-corporeal circulation is considered a safe and routinary technique. For this kind of procedure heart-lung machines (HLM) are used in the operating room. These systems are operated by perfusionists in cooperation

with anesthesiologists and heart surgeons who constantly monitor the patient and do adjustments when needed. The HLM main components consist of several roller pumps, a large reservoir and an oxygenator. The conventional HLM however is not suitable to be used for the purpose of extracorporeal circulation outside the hospital since the equipment is considerably large and heavy impeding its easy transportation.

Recent development of smaller and more efficient centrifugal pumps with miniaturized monitoring systems and improved oxygenators have allowed the construction of portable devices that can be easily taken outside the hospital. One example of such a device is the Lifebridge B₂T System. It is a portable, modular and easy-to-use system, designed to be applied in emergency transportation or resuscitation [19]. In this thesis this type of device is referred to as a portable extracorporeal circulatory support system (ECSS) to be differentiated from the conventional HLM.

The smaller size of the ECSS allows these systems to be transported in ambulances or helicopters to be taken directly where the patient is located. These devices however still need constant supervision and adjustments similar to the conventional HLM. In emergency situations trained perfusionists will not always be available on site and while the patient is being transported complications may arise. Under these circumstances quick decisions are needed together with proper adjustments to the system to assure the patients survival. The integration of a control mechanism can help by constantly monitoring the patient and upon these readings adjust the ECSS to provide the patient with optimal blood perfusion. This may reduce the workload of the operator and decrease the possibility of making incorrect decisions.

2. Objective

The objective of this thesis is to create a control mechanism for the ECSS that enables the automation of such a device. The task of the controller is to provide the patient with optimal organ perfusion and allow paramedics to focus on the safe transportation of the patient while working together with the machine.

The design of such control mechanism is of high complexity. The settings of the system may be different from one patient to another depending on medical preconditions and specific patient characteristics. As mentioned previously, the ECSS may assist the heart in perfusing the body, or in other cases the heart may be at full stop and the ECSS should be able to take full charge of perfusion. In any of these circumstances the system should provide with sufficient blood flow and pressure to reach all the organs; The blood stream going into the body should also be saturated with oxygen and without carbon dioxide.

In the operating room trained perfusionists constantly monitor the patients vital signals and, based on their knowledge and expertise they do proper adjustments to the HLM. The designed controller should be able to take similar decisions by using this knowledge and expressing them into control actions. Taking this into consideration fuzzy logic was chosen as a control mechanism. This allows the possibility to create rule sets based on the experts knowledge which are translated into actions by the fuzzy controller depending on the given inputs. In return it may give the operator an understanding of why certain changes are being made by analyzing the active rules at a specific point in time.

Additionally an adaptive mechanism is intended to make further adjustments to the

rules created improving control performance, adapting to the specific needs of each individual patient.

The extent of this work is not only to provide a solution for the automation of the ECSS but also to serve as ground work for the development of automation systems of other medical devices with the use of adaptive fuzzy controllers. With this in mind a software application framework was created with the purpose of providing the necessary components for automation. To comply with portability requirements a small embedded system was adapted to contain the developed controllers. The embedded system has low power consumption and the required sensors may be easily connected.

3. Methodology

Before the control design could take place several steps were followed which describe the layout of the thesis. First experiments using an animal model were used to obtain the characteristics of how the cardiovascular system works under extracorporeal circulation (ECC). Several sensors such as pressure and flow were used to obtain vital signals. An acquisition system was designed to capture and record all of the sensor generated data. Additionally an experimental protocol was created to identify the different steps of the procedure which could be afterwards analyzed and compared. Among other things, these experiments permitted the acquisition of data related to the interaction between the flow produced by the beating heart and the counter flow induced by the centrifugal blood pump through the femoral artery.

A mathematical model was designed describing the cardiovascular system together with the ECSS. The model was kept simple while still being able to represent physiological behavior. A gas exchange model was included, for oxygen delivery and carbon dioxide removal, together with a medication model for the administration of vasoactive drugs. The experimental data was used to validate the mathematical model and adjust its different parameters. With this mathematical model a simulation environment was created enabling continuous simulation where different control mechanisms could be tested and compared.

Constant communication with hospital perfusionists took place in order to create rules that could be used by the fuzzy controller. This was subdivided into different controllers in charge of modifying specific settings of the ECCS. The variables considered for controlling were centrifugal pump speed, gas blender parameters such as fraction of inspired oxygen (FiO₂) and gas flow, and the administration of medication such as vasoconstrictors and vasodilators.

An adaptive mechanism was designed to modify existing rules in order to improve the control performance depending on the specific needs of each patient. Several test cases were created to simulate different patient scenarios and evaluate control performance.

The software application framework was designed focusing on the easy design and implementation of the fuzzy controller. This framework allows the description of the control architecture through XML configuration files, eliminating the need of recompilation when changes are made to the system.

As a final step the previous knowledge and components are connected together; Using the simulation environment several scenarios were created; The ECSS automation was developed using the medical knowledge and implementing the fuzzy controllers with the

created software framework. The simulation and control are then joined for the control evaluation.

4. Thesis Overview

The following is a short summary of each chapter presented in this thesis also shown in figure I-1:

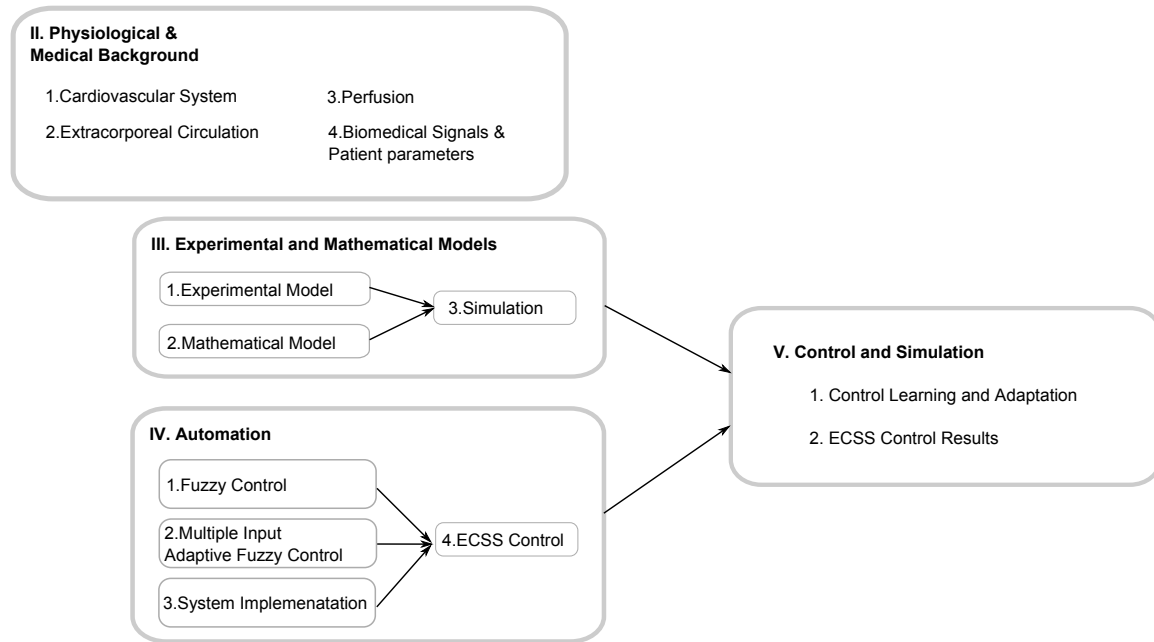


Figure I-1.: Thesis Overview

Chapter II: Physiological and Medical Background

This chapter gives the fundamental medical and physiological aspects necessary for this work. First the human cardiovascular system is explained together with the principals of gas exchange occurring during respiration (II.1). This is followed a section covering the extracorporeal circulation and a brief history of extracorporeal circulation with a description of its main components (II.2).After this a section is presented describing the principles of perfusion (II.3). The last section of this chapter will describe the different biomedical signals that are relevant for this work, how they are captured using different sensors and the typical patient parameter values obtained through these signals(II.4).

Chapter III: Experimental and Mathematical Models

This chapter contains the experimental and mathematical models. First the procedure of the animal experiment is presented, including the operation protocol and how the data

from the different vital signals was captured (III.1) Afterwards the mathematical model is described consisting on several sub models including the cardiovascular system, gas exchange and medication model. The data from the animal experiments is then used to adjust the different parameters of the mathematical model (III.2). This chapter will also include the developed simulation environment used to contain the mathematical model and allow continuous simulations(III.3). A comparison of both models is presented in the last section of this chapter.

Chapter IV: Automation of the Extracorporeal Circulatory Support System

This chapter describes the automation of the ECSS by using adaptive fuzzy controllers. This starts with an introduction to fuzzy logic(IV.1), followed with the description of a multiple input adaptive fuzzy control mechanism(IV.2). The next section describes the software framework designed to contain the fuzzy controller together with the necessary components for the automation of a medical device. This includes the adaptation of the framework into an embedded system (IV.3). The knowledge of the fuzzy control together with the software framework is then applied to the ECSS automation by creating three controllers, one in charge of adjusting the pump speed, a second to control the gas blender and a third used to deliver medication (IV.4).

Chapter V: Control and Simulation Results

This chapter shows the simulation results after connecting the simulation environment developed in chapter III together with the control system that is mentioned in chapter IV. First several control techniques are compared to test a learning and adaptive mechanism (V.1). Afterwards different test scenarios are presented to evaluate the ability of the designed controllers to provide a good perfusion (V.2).

Chapter VI: Discussion and Conclusions

This last chapter discusses the work of this thesis, considering the presented results and goals achieved (VI.1). At the end a final conclusion is given (VI.2) together with possible areas of future work including possible improvements and additional applications where the presented work may be applied (VI.3).

Appendix

The appendix is subdivided into several sections, first listing the used abbreviations and software tools (A). A second section gives a more detailed description of the developed software used for data acquisition, processing and the further details of the application framework used for control together with the developed simulation (B). Then the configuration files created for the ECSS and introduced into the software framework are shown (D) and at the end the complete mathematical model is presented (C).

Physiological and Medical Background

Contents

1. Cardiovascular System	7
2. Extracorporeal Circulation	13
3. Perfusion	20
4. Biomedical Signals and Patient Parameters	26

1. Cardiovascular System

The heart and circulatory system (also called cardiovascular system) make up a network that delivers blood to the body's tissues. With each heartbeat, blood is sent throughout the body, carrying oxygen and nutrients to all of the cells. Additionally wastes such as carbon dioxide produced by the body cells are transported and later on extracted by different organs in the body, being the lungs in charge of getting rid of the carbon dioxide and providing oxygen.

The circulatory system is composed of the heart and blood vessels, including arteries, veins and capillaries. It can be divided into two circulatory systems: The pulmonary circulation consisting of a short loop from the heart to the lungs and back again and the systemic circulation that sends blood from the heart to all of the parts of the body and back.

1.1. Heart

The heart is the key organ in the circulatory system. Its main function is to propel blood throughout out the body. It usually beats from 60 to 100 times per minute. The heart has four chambers that are enclosed by thick, muscular walls. The bottom part of the heart is divided into two chambers called the right and left ventricles, which pump blood out of the heart. A wall called inter ventricular septum divides the ventricles. The upper part of the heart is made up of the other two chambers called right and left atria. These chambers receive the blood entering the heart. A wall called inter atrial septum divides the atria, and they are separated from the ventricles by the atrioventricular valves. The tricuspid valve

separates the right atrium from the right ventricle, and the mitral valve separates the left atrium and the left ventricle.

Two other heart valves separate the ventricles and the large blood vessels that carry the blood leaving the heart. These valves are called the pulmonic valve, which separates the right ventricle from the pulmonary artery leading to the lungs, and the aortic valve, which separates the left ventricle from the aorta, the body's largest blood vessel.

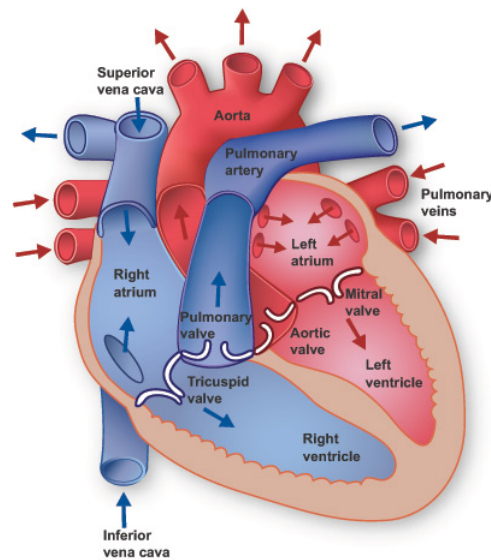


Figure II-1.: Heart anatomy [1].

1.2. Systemic Circulation

Blood vessels carrying blood away from the heart are called arteries. They are the thickest blood vessels, with muscular walls that contract to keep the blood moving away from the heart and through the body. In the systemic circulation, oxygen-rich blood is pumped from the heart into the aorta. The first section of this artery coming up and out of the heart is called ascending aorta. Two coronary arteries branch off at the beginning of the ascending aorta and divide into a network of smaller arteries that provide oxygen and nourishment to the muscles of the heart. This artery then curves forming the arch of the aorta. Here several arteries form branches to transport blood to the upper body (head, arms). After the arch of the aorta starts the descending aorta. From this point blood is taken to the rest of the body in several ramifications. From the right ventricle, the pulmonary artery divides into right and left branches, on the way to the lungs where blood is filled with oxygen.

Blood vessels that carry blood back to the heart are called veins. They are not as muscular as arteries, but they contain valves that prevent blood from flowing backward. The two largest veins are the superior and inferior vena cava. A network of thin capillaries connects the arteries and veins. Through them nutrients and oxygen are delivered to the cells. In addition, waste products such as carbon dioxide are removed from the cells also in the capillaries and introduced into the blood stream.

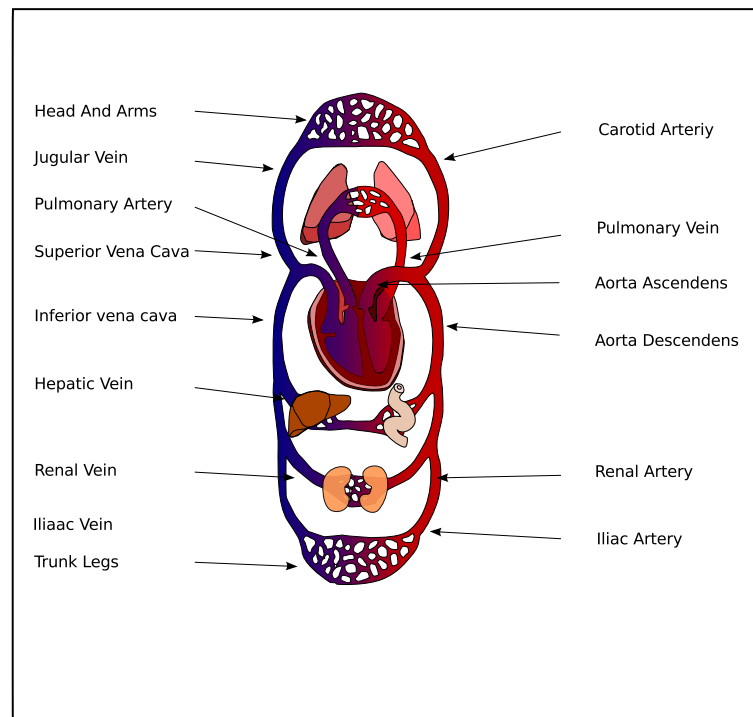


Figure II-2.: Cardiovascular System

1.3. Cardiac Cycle

A complete heartbeat makes up a cardiac cycle consisting of two phases. In the first phase, the ventricles contract (systole), sending blood into the pulmonary and systemic circulation. To prevent the flow of blood backwards into the atria during systole, the atrioventricular valves close. When the ventricles finish contracting, the aortic and pulmonic valves close to prevent blood from flowing back into the ventricles. Then the ventricles relax (diastole) and fill with blood from the atria, which makes up the second phase of the cardiac cycle.

A unique electrical system in the heart causes it to beat in its regular rhythm. The sinoatrial or SA node, a small area of tissue in the wall of the right atrium, sends out an electrical signal to start the contracting of the heart muscle. These electrical impulses cause the atria to contract first; they then travel down to the atrioventricular or AV node, which acts as a kind of relay station. From here the electrical signal travels through the right and left ventricles, causing them to contract and force blood out into the major arteries.

In the systemic circulation, blood travels out of the left ventricle, to the aorta, to every organ and tissue in the body, and then back to the right atrium. The arteries, capillaries, and veins of the systemic circulatory system are the channels through which this long journey takes place. Once in the arteries, blood flows to smaller arterioles and then to capillaries.

1.4. Gas Exchange

Gases are molecules that are free to move among one another in a process called “diffusion”. This also applies for gases dissolved in the fluids and tissues of the body. Except at absolute zero temperature, all molecules are continually undergoing motion.

Partial Pressures

Blood contains a mixture of gases, mainly oxygen, nitrogen, and carbon dioxide. The rate of diffusion of each of these gases is directly proportional to the pressure caused by that gas alone, which is called the partial pressure of that gas. The partial pressures of each gas are referred to as P_{O_2} , P_{CO_2} and P_{N_2} . Within the body gas exchange occurs in the capillaries.

1.4.a. Oxygen Delivery

Additionally to the oxygen diffused in the blood red blood cells contain a hemoglobin protein which has an oxygen binding capacity increasing the capability of blood to transport oxygen.

While in the capillaries, the bloodstream delivers oxygen and nutrients to the body's cells and picks up waste materials. Blood then goes back through the capillaries into venules, and then to larger veins until it reaches the vena cava. Blood from the head and arms returns to the heart through the superior vena cava, and blood from the lower parts of the body returns through the inferior vena cava. Both vena cava deliver this oxygen-depleted blood into the right atrium. From here the blood exits to fill the right ventricle, ready to be pumped into the pulmonary circulation for more oxygen.

In the pulmonary circulation, blood low in oxygen but high in carbon dioxide is pumped out the right ventricle into the pulmonary artery, which branches off in two directions. The right branch goes to the right lung, and vice versa. In the lungs, the branches divide further into capillaries. Blood flows more slowly through these tiny vessels, allowing time for gases to be exchanged between the capillary walls and the millions of alveoli which are tiny air sacs in the lung.

During the process called oxygenation, oxygen is taken up by the blood stream. Oxygen locks onto the hemoglobin contained in the red blood cells. The newly oxygenated blood leaves the lungs through the pulmonary veins and heads back to the heart. It enters the heart in the left atrium, then fills the left ventricle so it can be pumped into the systemic circulation.

1.4.b. Carbon Dioxide Removal

Carbon dioxide is continually being formed in the body by the different intracellular metabolic processes, the carbon in the foods being oxidized by oxygen to form carbon dioxide. This in turn diffuses into the interstitial fluids and blood, and is transported to the lungs where it diffuses into the alveoli and is transferred to the atmosphere by pulmonary ventilation. Several minutes are required for this passage of carbon dioxide from the cells to the atmosphere. If the rate of pulmonary ventilation is increased, the rate of expiration of

carbon dioxide becomes increased, and this decreases the amount of accumulated carbon dioxide in the extracellular fluids.

1.4.c. Respiratory Regulation of Acid-Base balance

An increase in carbon dioxide concentration in the body fluids lowers the pH toward the acidic side, whereas a decrease in carbon dioxide raises the pH toward the alkaline side. It is on the basis of this effect that the respiratory system is capable of altering the pH either up or down.

Effects of acidosis on the body There are two different kinds of acidosis: respiratory and metabolic acidosis. The major effect of acidosis is depression of the central nervous system. When the pH of the blood falls below 7.0, the nervous system becomes so depressed that the person first becomes disoriented and, later, comatose. Therefore, patients dying of diabetic acidosis, uremic acidosis, and other types of acidosis usually die in a state of coma. In metabolic acidosis the high hydrogen ion concentration causes increased rate and depth of respiration caused by the body trying to get rid of carbon dioxide. Therefore, one of the diagnostic signs of metabolic acidosis is increased pulmonary ventilation. On the other hand, in respiratory acidosis, respiration is usually depressed, which is opposite to the effect in metabolic acidosis.

Effects of alkalosis on the body The major effect of alkalosis on the body is over excitability of the nervous system. This effect occurs both in the central nervous system and in the peripheral nerves, but usually the peripheral nerves are affected before the central nervous system. The nerves become so excitable that they automatically and repetitively fire even when they are not stimulated by normal stimuli. As a result, the muscles go into a state of tetany, which means a state of tonic spasm. This tetany usually appears first in the muscles of the forearm, then spreads rapidly to the muscles of the face, and finally all over the body. Alkalotic patients may die from tetany of the respiratory muscles. Only occasionally does an alkalotic person develop severe symptoms of central nervous system over excitability. The symptoms may manifest themselves as extreme nervousness or, in susceptible persons, as convulsion. For instance, in persons who are predisposed to epileptic fits, simply over breathing often results in an attack. [7, p. 438]

1.5. Heart and Circulatory System Failures

Problems with the cardiovascular system are common, more than 64 million Americans have some type of cardiac problem. The following are the most common disorders.

1.5.a. Coronary Artery Disease.

Coronary artery disease is the most common heart disorder in adults, and it's caused by atherosclerosis. Deposits of fat, calcium, and dead cells form on the inner walls and clog up the body's arteries and get in the way of the smooth flow of blood. When the coronary arteries (blood vessels that supply the heart) get blocked a heart attack may be produced.

Hypercholesterolemia (high cholesterol). Cholesterol is a waxy substance that is found in the body's cells, in the blood, and in some of the foods we eat. Having too much cholesterol in the blood, also known as hypercholesterolemia or hyperlipidemia, is a major risk factor for heart disease and can lead to a heart attack.

Hypertension (high blood pressure) is when a person has blood pressure that's significantly higher than normal. Over time, it can cause damage to the heart and arteries and other body organs. Teens can have high blood pressure, which may be caused by genetic factors, excess body weight, diet, lack of exercise, and diseases such as heart disease or kidney disease.

1.5.b. Cardiogenic Shock

Cardiogenic shock is a major, and frequently fatal, complication of several acute and chronic disorders that impair the ability of the heart to maintain adequate tissue perfusion.

Causes Myocardial infarction is the most common cause of cardiogenic shock in modern times. It is the interruption of blood supply to a part of the heart, causing heart cells to die. This is most commonly due to occlusion (blockage) of a coronary artery. The resulting ischemia (restriction in blood supply) and oxygen shortage, if left untreated for a sufficient period of time, can cause damage or death (infarction) of heart muscle tissue (myocardium).

Cardiovascular mechanics Under cardiogenic shock the left ventricular end-systolic pressure-volume curve is shifted to the right due to a severe reduction in contractility. At a similar or lower systolic pressure the ventricle ejects less blood volume per beat. This causes the end-systolic volume to increase and the stroke volume is decreased.

1.5.c. Mitral Valve Regurgitation

Mitral valve regurgitation happens when the heart's mitral valve does not close tightly, which allows blood to flow backward in your heart. When the mitral valve does not work properly, blood can't move through the heart or to the rest of the body as efficiently, producing fatigue. Treatment of mitral valve regurgitation depends on how severe your condition is, whether it's getting worse, and signs and symptoms. For mild cases, treatment may not be necessary. Heart surgery may be needed to repair or replace the valve for more-severe cases. Left untreated, severe mitral valve regurgitation can cause heart failure or serious heart rhythm problems (arrhythmias).

2. Extracorporeal Circulation

Extracorporeal circulation (ECC) is a medical procedure in which blood is taken outside the body and processed through an external device before it is introduced back into the body. In the case of the extracorporeal circulatory support system (ECSS) the blood is taken out of the body for carbon dioxide-oxygen exchange and pumped back into the body to replace or complement the action of the heart and lungs.

For a person suffering from cardiogenic shock, cardio pulmonary resuscitation may not be sufficient to restore normal heart function due to a possible obstruction or valve malfunction. The heart may still work but not with enough strength to provide sufficient cardiac output to perfuse the whole body. For these cases surgical intervention is needed, which is at times hours away from where the patient is located. Extracorporeal circulation through a portable ECSS may provide a higher chance of survival to the patient by giving the body proper perfusion while being transported to the hospital. Once in the operation room the ECSS may facilitate the process of intervention and provide of a better outcome.

This section will first give a brief history of extracorporeal circulation. Then the ECSS will be described in detail, including its main components. The final section will describe the perfusion procedure performed during ECC together with the patient's parameters that are normally observed.

2.1. History

In the year of 1930 Dr John Gibbon, Jr. thought the idea of creating a machine that could take over the functions of the heart and lungs, enabling the possibility of having surgeries on the heart itself while the rest of the body could be perfused with oxygen rich blood and remove the carbon dioxide. This would eventually become the beginning of the cardiopulmonary bypass and the creation of the hear-lung machine. It took 23 years to come from the idea to the realization of the first extracorporeal system. Since then numerous designs of the heart-lung machine (HLM) have been created. In 1946 Gibbon started working together with Thomas Watson from IBM to create the first HLM called Model I. This machine was successful in animal experiments however it was too heavy to be transported and used with humans. This brought to the development of Model II which was finished by 1952. This machine however did not bring any success when used in humans and stalled the use of the HLM in the operating room. Serious problems of clotting and blood loss had to be solved.

Stuckey [20] in 1958 reported a series of three patients treated with ECC for cardiogenic shock following acute myocardial infarction. Despite criteria requiring patients to be moribund before inclusion, one patient survived to hospital discharge.

2.1.a. Evolution of the HLM

With the expanded possibilities of uses of extracorporeal circulation the heart lung machine has evolved to different types of devices. In the operating room heart lung machines are used for open heart procedures where the heart and lungs are directly bypassed while the surgeon operates. In this case an anesthesiologist and a perfusionist constantly monitors the patients vital signs, where the first is in charge of providing the needed anesthesia

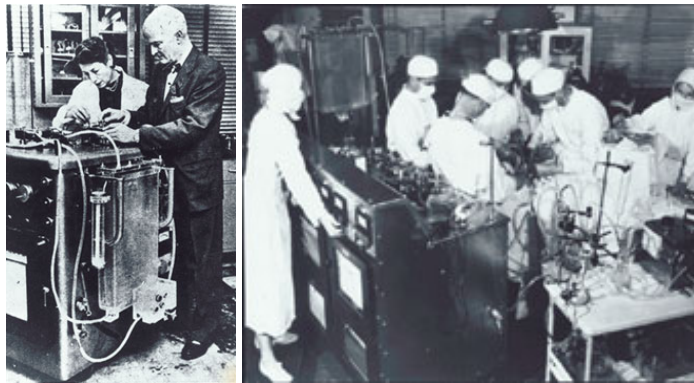


Figure II-3.: Gibbon creator of the first Heart-Lung Machine

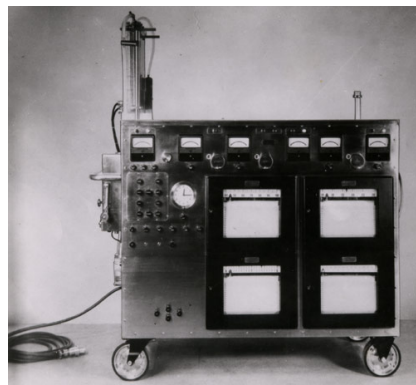


Figure II-4.: Model II Manufactured by IBM

and medication and the second adjusts the parameters of the HLM and gas exchange. The HLM used in this modality is composed of several roller pumps, where one may be spared in case of malfunction of the other. A big reservoir is used to retain the blood of the patient and an oxygenator with a heat exchanger is used to oxygenate the blood and maintain adequate temperature, in some cases causing hypothermia for better perfusion. Another type of system is called an extracorporeal membrane oxygenation or ECMO, this type of device does not strictly bypass the heart and lungs, but is used as an assistant device for patients that are stationed in an intensive care unit (ICU) and that may have insufficient lung or heart function. The time of use for these type of devices is from three to ten days where as for the previous device the operation time is less than a day. In this case the ECMO device is operated by an ECMO clinical specialist. The objective of this type of machine is to keep the patient's vital signals in a stable state. A monitoring system checks the vital signs of the patient and proper adjustments are done manually by the specialist. The flow rates of these types of devices are in the range of 70 to 80 % of the cardiac output. The cannulation is done via the neck.

The third type of extracorporeal circulation system is a device that is meant for resuscitation purposes. The patient may be outside the hospital or inside but may not be under direct observation. In this case the patient may be suffering cardiogenic shock / myocar-

dial infraction. As seen in the previous chapter different causes may be related to this condition. In case of cardiogenic shock the heart is at a full stop, The normal procedure is to first try to resuscitate the patient with cardiopulmonary resuscitation, however when this procedure is not effective a portable heart lung machine or extracorporeal circulatory support system may be used. Previous research has shown that continuous tries of CPR may bring greater complications and reduce the possibility of survival[21]. With the use of an ECSS the heart may remain inactive while the rest of the body may be adequately perfused. This may give the paramedics enough time to bring the patient to the hospital where proper surgery may take place. This type of device has additional requirements such as portability and ease of use.

Figure II.5(a) shows a modern heart-lung machine used in the operating room. It consists of several roller pumps and extensive sensors to monitor the system. Figure shows an ECMO from Sorin consisting of an oxygenator with an integrated temperature exchange. Figure II.5(c) and II.5(d) shows a mini-heart-lung machine from Maquet, and an extracorporeal mechanical support system from Lifebridge respectively.



(a) Sorin SIII heart-lung machine.



(b) EOS Sorin ECMO



(c) Maquet mini-heart-lung machine.



(d) Lifebridge extracorporeal mechanical support system.

Figure II-5.: Different machines for extracorporeal circulation.

2.2. ECC for Cardiac Resuscitation and Life Support

For the case of extracorporeal circulation for cardiac resuscitation and life support the portable ECSS is connected to the patient through femoral cannulation. This is shown in figure II-6.

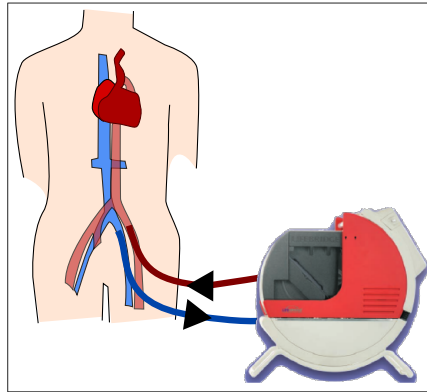


Figure II-6.: Connection of ECSS with patient

2.3. Components of Extracorporeal Circulation

The main components used for extracorporeal circulation are depicted in figure II-7. They consist of a reservoir, a blood pump, an oxygenator, an arterial filter and arterial and venous cannula. A gas blender is used to regulate the mix of oxygen-air that is provided to the oxygenator. Additional sensors such as pressure, flow and bubble detectors are used to adjust the settings of the ECSS and assure of a proper operation.

2.3.a. Reservoir

A reservoir is placed immediately before the systemic pump to serve as a holding tank and act as a buffer for fluctuation and imbalances between venous return and arterial flow. The reservoir also serves as a bubble trap for air that enters the venous line. Additionally it provides time for the perfusionists to act if venous drainage is reduced dramatically. In a portable system the reservoir is of less capacity than in a normal heart lung machine, consisting of 1 to 2 liters.

2.3.b. Blood Pump

The blood pump is in charge of circulating the blood out of the body and through an oxygenator. There may be several types of pumps, however they should be able to move large volumes of blood (7 L/m) against significant pressures (500 mm Hg). The handling of blood by the pump should minimize flow velocity so that damage to the blood is minimized. Pump components in contact with blood should not damage the blood cells and should not activate either the inflammatory or the coagulation cascades. The pump de-

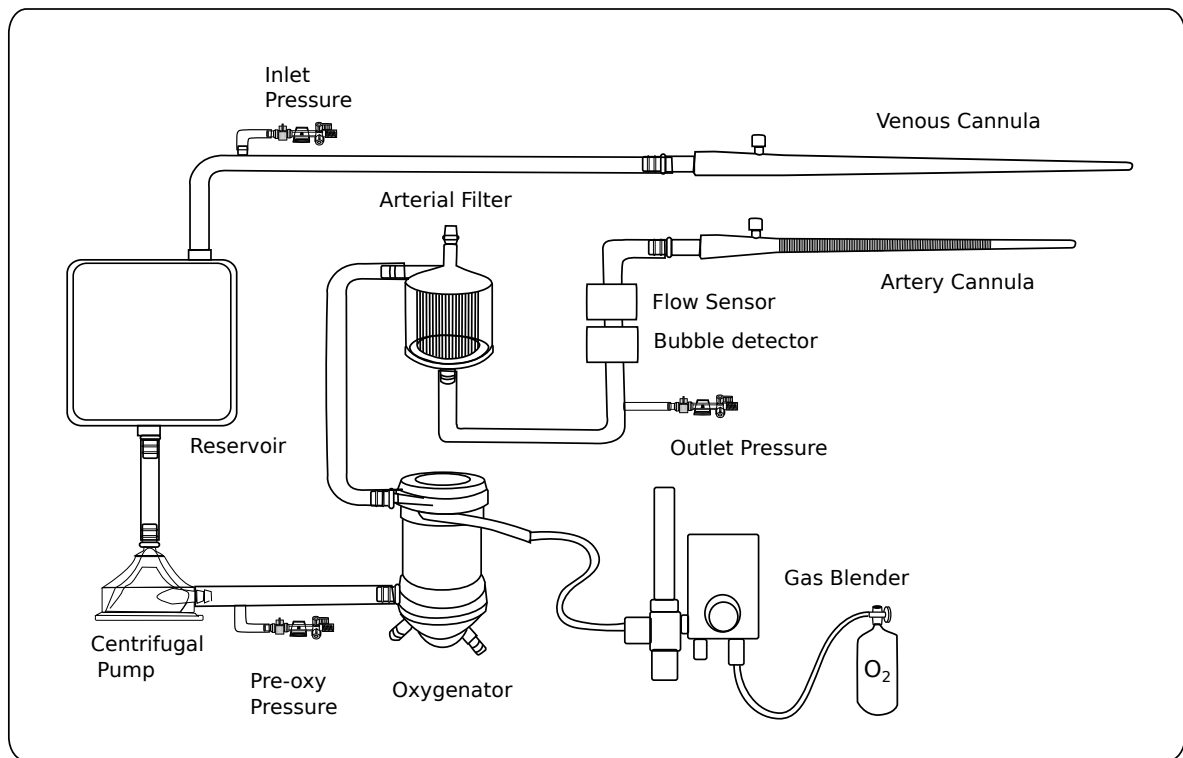


Figure II-7.: Extracorporeal Support System components

sign should be simple and free from dead spaces and recesses to avoid stagnation and turbulence.

There are two main types of pumps: displacement pumps and rotary pumps. The typical displacement pump is the roller pump. It consists of two rollers placed opposite to each other which move the blood through a piece of tubing by occlusion. This pump is capable of generating both positive and negative pressure, being possible to be used for suction (Figure II.8(a)).

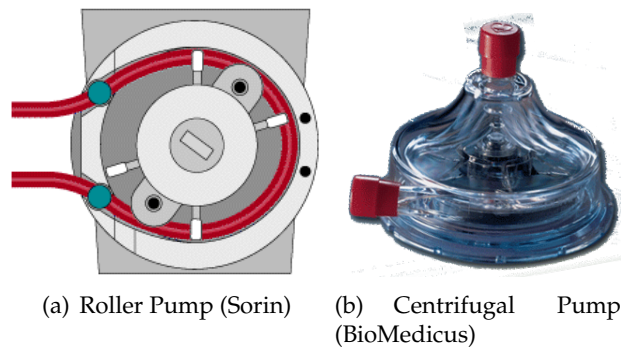


Figure II-8.: Types of blood Pumps

Rotary pumps are composed of a smooth plastic cone located inside a plastic hous-

ing. They move fluids by creating a pressure gradient between the inlet and the outlet of the pump (Figure II.8(b))[22]. By rotating the inner cone rapidly (2000-4000 rpm) these pumps generate a pressure differential that causes the movement of fluid. This pressure gradient results from the vortex created by the pump head. The rotary motion creates an area of low pressure in the center and an area of high pressure on the sides. The resulting rate of blood flow depends on the pressure gradient and the resistance at the outlet of the pump. The resistance at the outlet of the pump is a function of two components: the CPB circuit (oxygenator, filter, tubing, arterial and venous cannula) and the SVR of the patient.

Some of the advantages of centrifugal pumps over roller pumps is the lower risk of pumping massive air emboli into the arterial line. They become de-primed and stop pumping if more than approximately 50 ml of air is introduced into the circuit. They may still however pass small quantities of smaller bubbles which may be extracted by a filter. When a centrifugal pump is not running and is connected to the patient's arterial system blood may flow backward through the pump and out of the patient, for this reason the systemic line is clamped before being completely stopped. Additionally if the arterial line becomes occluded, the centrifugal pump will not generate excessive pressure (max 700-900 mmHg) compared to the occluding roller pump. This may prevent the rupture of the systemic flow line. In terms of negative pressure it will be limited to approx -500 mmHg reducing the risk of cavitation and microembolus pressure.

2.3.c. Oxygenator and Gas Blender

During ECC the oxygenator acts as the artificial lung. The fundamental purpose of the oxygenator is to remove the excess carbon dioxide and oxygenate the blood by increasing the partial pressure. (PO_2).

Gas transfer in oxygenators occurs due to diffusion. This consists of the random movement of molecules from the diffusing gas from regions of higher concentrations to regions of lower concentrations. The rate of diffusion is proportional to the partial pressure gradient of the gas in the direction of diffusion.

The most common type of oxygenators used are membrane oxygenator. In these types of oxygenators the gas flow is separated from the blood-flow. The separating membrane is a semi-permeable membrane, consisting of polypropylene or silicon rubber.

The oxygenator is connected to a gas blender that provides a mixture of oxygen and air. This mixture is regulated by adjusting the fraction of inspired oxygen (FiO_2) and the amount of flow.

2.3.d. Filter and Bubble Trap

Micro pore filters are used to eliminate gross and microembolic gas (GME) bubbles and other micro particles that may be present in the ECC circuit. Micro pore filters may be used in several locations within the extracorporeal circuit, being the most common place in the arterial line after the oxygenator.

2.3.e. Sensors

The ECSS contains sensors that give information about the status of the machine and the patient.

Pressure sensors The pressure sensors included in the ECSS are inlet pressure located at the inlet of the reservoir, oxygenator pressure sensor located between the centrifugal pump and the oxygenator and outlet pressure located after the arterial filter.

Flow sensor and bubble detector An ultrasonic flow probe is used to detect the flow generated by the ECSS located at the arterial line. A bubble sensor is also used to detect for any possible air bubble after the arterial filter. In the case of the Lifebridge system when a bubble is detected a fast arterial clamp is activated and blood is recirculated in the machine until air elimination is guaranteed.

Additionally a low-level sensor is placed inside the reservoir to detect critical level that may cause air to be introduced into the blood pump.

2.3.f. Cannula

The cannulas are the connection between the patient and the ECSS. For the case of the portable ECSS femoral cannulation is applied. Many different types of cannulas made of various materials are available.

Venous cannula The venous cannula is inserted in the femoral vein. This tubing collects the blood low of oxygen from the patient, and into the reservoir. Venous drainage may be accomplished by gravity siphonage. However in a portable ECSS this is done by applying suction, eliminating the need of having the system bellow the patient. Excessive drainage may cause the compliant vein walls to collapse around the ends of the venous cannulas and intermittent reduction of venous drainage. This risk may be reduced by partially occluding the clamp on the venous line.

Arterial cannula The arterial cannula is usually the narrowest part of the ECSS. High flow through narrow cannulas may lead to high pressure gradients, high velocity of flow, turbulence, and cavitation. Hemodynamic evaluations of arterial cannulas have traditionally been based on measurement of the pressure drop. Pressure gradients exceeding 100 mmHg are associated with excessive hemolysis and protein denaturation. For this reason it is important to select a cannula that provides adequate flow with no more than 100 mmHg pressure gradient.

3. Perfusion

Perfusion may be considered minimally acceptable if the patient survives without life-threatening complications or persistent clinically manifest organ dysfunction. An optimal perfusion may be defined as the best long-term patient outcome in terms of survival and function of all organ systems. Optimal perfusion should be associated with minimal activation of inflammation, coagulation and of the autonomic and endocrine systems, preservation of homeostasis and oncotic pressure, the least morbidity and disturbance of organ function, and the fast recovery. [24]

There is however not a specific procedure that may apply to all cases and extensive research has been done on trying to find the best perfusion technique. [24–29]

The primary determinants of tissue oxygen supply and demand include mean arterial blood pressure, extracorporeal flow rates, type of flow (pulsatile versus non pulsatile), hematocrit values, systemic oxygen delivery (DO_2), temperature, and acid-base management. The following sections will describe these parameters and the values that may be adequate for perfusion.

3.1. Blood Flow

Suggested flow rates are calculated based upon body surface area and temperature management strategies. Most common flow rates during ECC are $2.2\text{--}2.5L/min/m^2$. The body surface area is taken into consideration. However lower flow rates of $1.2L/min/m^2$ during hypothermic bypass have been used by some investigators with good clinical results[30].

Systemic flow rates may impact perfusion of other organ systems besides the brain. In a swine model study [31], reductions in pump flow did not affect cerebral blood flow, but significantly reduced perfusion of all visceral organs. Increasing the pump flow restored perfusion to other organs, whereas restoration of systemic pressures with phenylephrine did not. On a similar investigation systemic pressures of 45 mmHg were generated with reduced pump flow. Perfusion of kidneys, gastrointestinal tract and pancreas was significantly reduced at this flow. From this studies it is suggested that blood flow to visceral organs may be compromised at lower pump flow rates.

Factors determining minimal safe EFR:

- Body Surface Area (BSA)
- Degree of hypothermia
- Acid-base balance
- Whole-body oxygen consumption
- Degree of neuromuscular blockade
- Oxygen content of blood (hemoglobin concentration and saturation, PO_2)
- Depth of anesthesia
- Specific organ ischemic tolerance

3.2. Arterial Pressure

Systemic arterial blood pressure is a function of flow rate, blood viscosity (hematocrit), and vascular tone. Perfusion of the brain is normally protected by auto regulation, but auto regulation appears to be lost somewhere between 55 and 60 mmHg during CPB at moderate hypothermia and a hematocrit of 24% [24]. Cerebral blood flow may be sufficient at lower arterial pressures; however several studies have found a lower mortality rate when pressures are maintained near $70 \pm 7\text{mmHg}$. For older patients, who may have vascular disease and/or hypertension, mean arterial blood pressure is generally maintained between 70 and 80mmHg at 37°C . Higher pressures are undesirable since it may rupture sutures and create further complications.

To guarantee organ perfusion mean arterial pressure (MAP) should remain above 50 to 60 mmHg. Cerebral blood flow is preserved even if systemic flow is less than normal. However, there is a hierarchical reduction of flow to other organs as total systemic flow is progressively reduced. First skeletal muscle flow falls, then abdominal viscera and bowel, and finally kidneys.

Potential advantages of higher MAPs

- Enhanced tissue perfusion in high risk patients (hypertensive, diabetic and elderly)
- Improved collateral flow to tissues at risk of ischemia.
- Allows higher pump flow rates.

Potential advantages of lower MAPs

- Less trauma to blood elements
- Reduced risk of vessel rupture
- Permits the use of smaller venous and arterial cannula
- Enhanced myocardial protection (reduced collateral coronary blood flow)

3.3. Hemodilution and Hematocrit

A main reason for employing asanguinous primes was the reduced use of blood. Crystalloid primes facilitated the use of ECC as an emergency procedure by reducing the time and resources required to have a usable circuit available. Previous studies done by Gordon et al[32] showed that when hemodilution is used to decrease hematocrit and the flow rate is held constant, perfusion pressure decreases in direct proportion to the change in viscosity. Clinical studies have shown that hemodilution passively increases venous return, possibly associated with the marked increase in flow in the small vessels, especially the post capillary venules.

The hematocrit is the proportion, by volume, of the blood that consists of red blood cells. Generally accepted flow rate at 35 to 37°C and hematocrit of 25% is approximately $2.4\text{L}/\text{min}/\text{m}^2$ in deeply anesthetized and muscle-relaxed patients. Hemodilution reduces

blood oxygen content from approximately 20 mL/dl to 10 to 12mL/dL; for this reason flow rates must increase over normal cardiac output or oxygen demand must decrease.

Hemodilution anemia may be produced when ECC is applied, caused by the use of asanguinous prime of circuits with conventional priming volumes. The degree of hemodilution anemia that is observed is related to the patients' initial red cell mass (body size and hematocrit) and priming volume of the ECSS. The advantages of hemodilution during ECC include reduced blood viscosity and improved micro circulatory flow. Excessive hemodilution, however may compromise oxygen delivery at the tissue level and contribute to hypotension.

The following is a list of common hematocrit levels according to age and sex:

- Newborns: 55-68%
- One (1) week of age: 47-65%
- One (1) month of age: 37-49%
- Three (3) months of age: 30-36%
- One (1) year of age: 29-41%
- Ten (10) years of age: 36-40%
- Adult males: 42-54%
- Adult women: 38-46%

3.4. Gas Exchange

Hemoglobin

Hemoglobin *Hb* is a protein in the red blood cells that is in charge of carrying oxygen from the respiratory organs to the rest of the body.

The hemoglobin oxygen saturation *HbO2Sat* is a quantitative measure of volume of oxygen per volume of blood, depending on the grams of hemoglobin per deciliter of blood. For example, two samples of blood, one with hemoglobin at 15 g/dl and the other at 7.5 g/dl, may both be 96% saturated with oxygen, but the hemoglobin oxygen saturation will be greater in the 15 g/dL sample.

Relation between Hematocrit and Hemoglobin concentration:

$$Hct[\%] = ((0.0485 \cdot ctHb[mmol/L] + 0.0083) \cdot 100) \quad (II.1)$$

or

$$ctHb = 1/3Hct \quad (II.2)$$

3.4.a. Oxygen Saturation

Oxygen is transported in the blood attached to hemoglobin molecules. Oxygen saturation is a measure of how much oxygen the blood is carrying as a percentage of the maximum of its capacity.

One hemoglobin molecule can carry a maximum of four molecules of oxygen, if a hemoglobin molecule is carrying three molecules of oxygen then it is carrying 3/4 or 75% of the maximum amount of oxygen it could carry.

One hundred hemoglobin molecules could together carry a maximum of 400 (100 x 4) oxygen molecules, if these 100 hemoglobin molecules were carrying 380 oxygen molecules they would be carrying $(380 / 400) \times 100 = 95\%$ of the maximum number of oxygen molecules that could carry and so together would be 95% saturated. Oxygen saturation is also referred to as SpO₂.

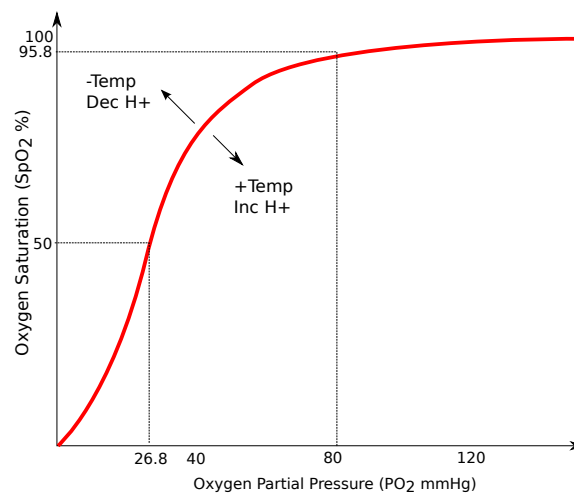


Figure II-9.: Relation of Hemoglobin Saturation and Oxygen Partial Pressure

3.4.b. Arterial PaO_2

PaO_2 should be kept above 150 mmHg to assure complete arterial saturation. It is not proven the effects of high levels of PaO_2 pressures. (above 200mmHg).

The aim is a $PACO_2$ of 40mmHg and pH of 7.4 measured at 37 degrees C.

Oxygen Delivery

Systemic Oxygen Delivery or DO_2 may be considered one of the most optimal determinants of "optimal" perfusion as mentioned by Glenn et al. It is the rate at which oxygen is transported from the lungs to the micro circulation. [24]. DO_2 is calculated by multiplying the pump flow rate by the arterial oxygen content:

$$DO_2 = Q \times ((Hb \times SpO_2 \times 1.36) + (0.003 \times aPO_2)) \quad (II.3)$$

DO_2 may be improved by increasing pump flows, increasing hematocrit concentrations or by increasing hemoglobin saturation at the amount of dissolved oxygen (increasing the inspired oxygen concentration [F_{IO_2}]). On a normal patient, assuming a cardiac index of 2.3 to 2.6 $L/min/m^2$, normal oxygen levels and a hemoglobin of 12 g/dL the value of DO_2 may be between 350-450 $mL/min/m^2$. During ECC hemoglobin may be decreased to 7 g/dL, reducing the DO_2 to 200-300 $mL/min/m^2$. The reduction is caused by a decrease in arterial oxygen content occurring from hemodilution once ECC has started. When the whole-body oxygen consumption is unchanged an increase in the oxygen extraction ratio is increased, proper adjustment of oxygen delivered to the oxygenator may be required. As DO_2 decreases, VO_2 initially remains stable due to increased tissue oxygen extraction. When maximal oxygen extraction is reached, whole body VO_2 and tissue oxygenation begin to decrease and metabolic acidosis begins to develop.

The normal heart-lung machines used for cardiopulmonary bypass in hospitals use larger extracorporeal circuits and reservoirs, increasing the amount of hemodilution, portable ECSS may require less priming solutions reducing the effect of lower hematocrit.

Oxygen Consumption

Oxygen consumption (VO_2) is the rate at which oxygen is removed from the blood for use by the tissues. Maximal oxygen consumption falls with decreasing flow as described in the following formula:

$$VO_2 = 0.44(Q - 62.7) + 71.6 \quad (II.4)$$

Normal VO_2 in a conscious, resting person is approximately 250 mL O_2 /min.

Oxygen Extraction

Oxygen extraction is the slope of the relationship between oxygen delivery (DO_2) and oxygen consumption (VO_2). It is most commonly expressed as the oxygen extraction ratio, which is the proportion of arterial oxygen that is removed from the blood as it passes through the micro circulation:

$$O_2ExtractionRatio = (CaO_2 - CvO_2)/CaO_2 \quad (II.5)$$

Normal oxygen extraction ratios range from 0.25 to 0.30.

3.5. Temperature

Hypothermia reduces oxygen consumption by a factor of 0.5 for every $10^\circ C$ decrease in temperature [27]. Hypothermia reduces overall VO_2 , however the balance between oxygen supply and demand can be impaired by reductions in tissue DO_2 due to increased blood viscosity, reduced micro circulatory flow, and a leftward shift of the oxygen-hemoglobin dissociation curve. There is no concrete proof showing that normothermic or hypothermic procedures will have better outcome than the other, and the way temperature is handled will depend mostly on the experience of the practitioner.

3.6. Anesthesia and Medication

During ECC anesthesia is used to maintain the patient in an unconscious and painless state, disconnecting certain reflex activities and the reversible paralysis of skeletal muscles. Haemodynamics, blood-gases, hypothermia, hemodilution among other parameters are monitored to predict the distribution and the effective duration of these substances in the body.

Additional drugs can be administered to help reach haemodynamic targets. Vasopressors may increase the coronary and cerebral blood flow during the low-flow state associated with shock. These drugs are called sympathomimetic since they mimic the effects of transmitter substances of the sympathetic nervous system. They may be of the type adrenergic alpha-agonist which is a drug that selectively stimulates alpha adrenergic receptors. These receptors directly affect the heart rate. Another type is the beta-adrenergic agonist which acts upon the beta receptors. From these type the β_1 may increase the heart and the atrial cardiac muscle contractility and the β_2 receptor may be use as a muscle relaxation agent.

Dopamine and dobutamine are drugs that may be used to improve cardiac contractility, with dopamine preferred for hypotensive patients.

Vasodilators relax vascular smooth muscle and reduce the systemic vascular resistance (SVR), allowing improved forward flow, which improves cardiac output.

Adequate pain control is essential for quality patient care and patient comfort.

If proper blood perfusion takes place, pH and blood-gas management regimens remain stable, and extracorporeal circulation may keep the functions of body organs unchanged while being in use. Decreased blood volume, caused by hemodilution and hypotension, hypothermia, haemolysis and anesthesia together with preexisting heart disease may result in tissue damage, temporary or permanent neurological damage and single or multiple organ failure.

4. Biomedical Signals and Patient Parameters

Physiological processes are complex phenomena, including nervous or hormonal stimulation and control. Most physiological processes are accompanied by or manifest themselves as signals that reflect their nature and activities. Such signals could be of many types, including biochemical in the form of hormones and neuro transmitters, electrical in the form of potential or current, and physical in the form of pressure or temperature. Diseases or defects in a biological system cause alterations in its normal, physiological processes, leading to pathological processes that affect the performance, health, and general well-being of the system. A pathological process is typically associated with signals that are different in some respects from the corresponding normal signals. By understanding a system of interest, in this case a patient, it is possible to observe the corresponding signals and assess the state of this patient. This section will describe different signals that may be obtained from the body, their characteristics and the type of sensors that may be used for its acquisition.

4.1. Blood Pressure

The cardiovascular system is a closed system, when the heart contracts its wall a high pressure is generated, producing as a consequence a blood flow from a high pressure compartment to a compartment with a lower pressure. From this effect two values of pressure may be measured. One is the systolic pressure produced during the contraction of the heart, and the diastolic pressure which occurs with the ventricular relaxation. Blood pressure is measured in millimeters of mercury (mmHg) in clinical practice, although the international standard unit for pressure is the Pascal. In intensive-care monitoring and surgical procedures, a pressure transducer may sometimes be inserted into an artery (along with other intra-arterial or intra-venous devices). It then becomes possible to obtain the arterial systolic and diastolic blood pressure on a continuous-time recording.

4.1.a. Pressure Transducers

Two types of sensors may be used to capture different pressures within the body. The first one is by using a disposable pressure transducer encapsulated in a plastic casing. This type of sensors are connected to a catheter. The second type of sensor uses the same principle however the size is reduced to fit on a thin wire (down to 2 Fr). These type of sensors may be introduced into the tubing and obtain more localized pressure curves. The sensor works by having small filaments call strain gauges that are in contact with the fluid. When these components are bent due to pressure the resistance will change, when supplied with a constant current the change of resistance generates a proportional change of voltage, which can be converted to equivalent pressure units of mmHg.

A calibration device such as the KAL 84 from Halstrup Walcher may be used to calibrate pressure transducers and obtain the equivalence of voltage to pressure units[10].

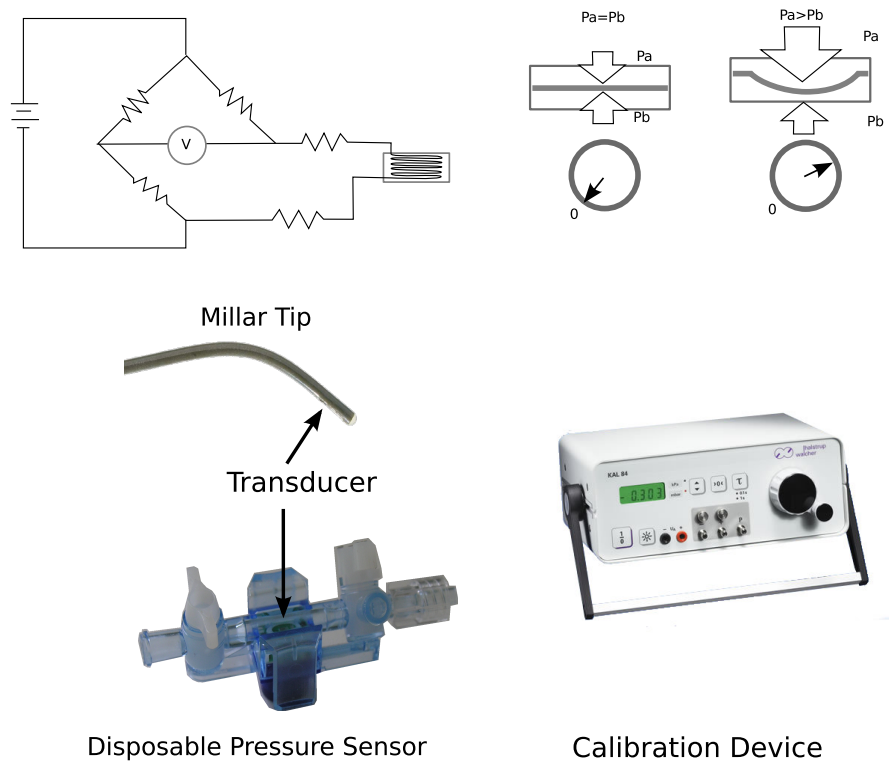


Figure II-10.: Strain gauge blood pressure transducer.

4.2. Blood Flow

The measurement of blood flow is important, since it shows how much perfusion is going into the body. This may be done by placing flow transducer around the artery or tube that is to be measured.

4.2.a. Flow Transducer

The flow transducers are based on an ultrasonic principle. They utilize the Doppler variance known as ultrasound transit-time. It consists of two small piezoelectric crystals, one upstream and one downstream, mounted in a common tip. The times are measured for an ultrasound pulse signal emitted from the upstream crystal to arrive at the downstream crystal through the reflector and for a signal from the downstream crystal to reach the upstream crystal through the reflector. Since ultrasound travels faster on the direction as the flow, a small time difference for the two signals expressed as a shift of phase is determined.

4.3. Heart Beat and Electrocardiogram

The electrocardiogram (ECG) is the electrical manifestation of the contractile activity of the heart, and can be recorded fairly easily with surface electrodes on the limbs or chest. The ECG is perhaps the most commonly known, recognized, and used biomedical signal.

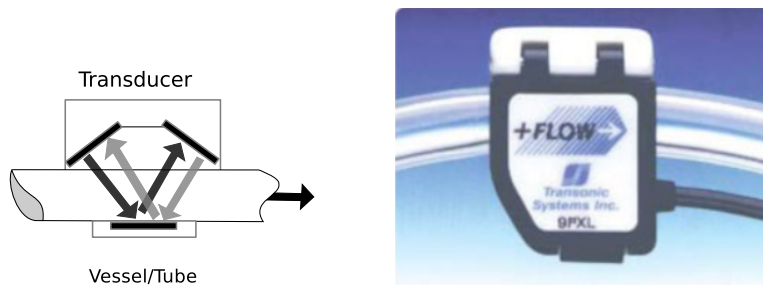


Figure II-11.: Blood flow transducer

The frequency of the heart in terms of beats per minute (bpm) may be easily estimated by counting the readily identifiable waves. The ECG wave shape is altered by cardiovascular diseases and abnormalities such as myocardial ischemia and infarction, ventricular hypertrophy, and conduction problems.

The heart rate (HR) or cardiac rhythm is controlled by specialized pacemaker cells that form the sino-atrial (SA) node located at the junction of the superior vena cava and the right atrium. The firing rate of the SA node is controlled by impulses from the autonomous and central nervous systems leading to the delivery of the neurotransmitters acetylcholine (for vagal stimulation, causing a reduction in heart rate) or epinephrine (for sympathetic stimulation, causing an increase in the heart rate). The normal (resting) heart rate is about 70 bpm. The heart rate is lower during sleep, but abnormally low heart rates below 50 bpm during activity could indicate a patient disorder. The instantaneous heart rate can reach values as high as 200 bpm during vigorous exercise or athletic activity; a high resting heart rate could be due to illness, disease, or cardiac abnormalities. Heart rates below 60 bpm are referred to as bradycardia whereas elevated heart rates are referred to as tachycardia. The upper threshold of a normal heart rate is dependent on the age of the patient.

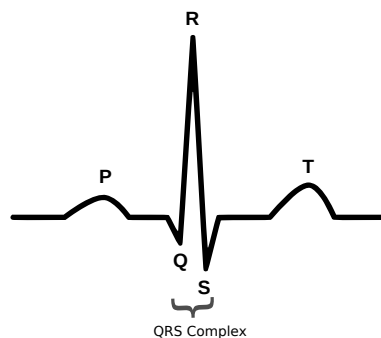


Figure II-12.: QRS Complex.

The QRS complex is one of the components of a typical electrocardiogram (ECG) tracing of a complete cardiac cycle, normally occurring after the P wave (atrial depolarization) and before the T wave (ventricular repolarization). It is a graphical representation of the cardiac impulse traveling into the ventricular myocardium.

About 120 to 200 milliseconds (ms) after the generation of an action potential in the SA node, the cardiac impulse reaches the ventricles through the atrioventricular (AV) nodal/His bundle system. The cardiac impulse is conducted to the ventricular mass through the bundle branches and the Purkinje fibers. The QRS complex represents ventricular depolarization. Normally, the QRS complex is seen as a 'sharp' complex, resulting from the rapid depolarization of the ventricles via the fast conducting Purkinje fibers of the bundle branches. Compared to the intra-atrial conduction speed, the conduction speed of the bundle branches is high, averaging about 4 meters per second.

The first deflection of the QRS complex is called a Q wave when it is negatively deflected (pointing downward) and an R wave when it is positively deflected (pointing upward). A positive deflection following a Q wave is called an R wave; and a negative deflection following an R wave is designated as an S wave. A second positive deflection after the R wave is called an R'.

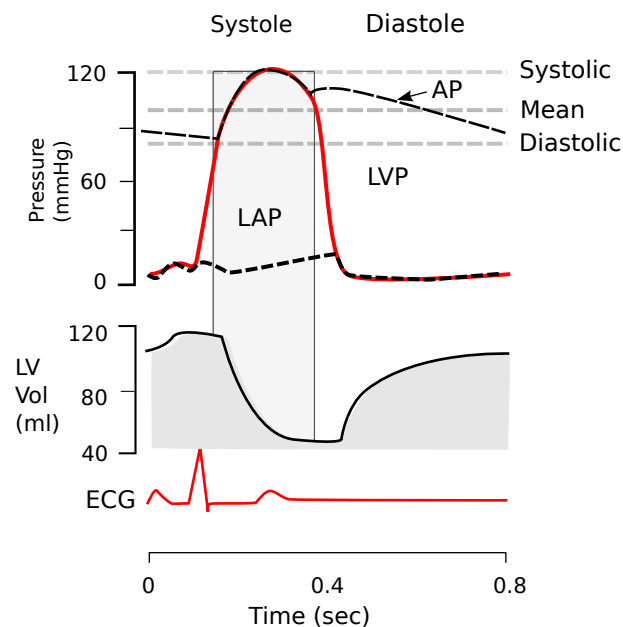


Figure II-13.: Pressure Volume and ECG curves

Figure II-13 show the relation between the electrocardiogram captured from the heart with volume from the left ventricle of the heart and pressures from the left ventricle (LVP), the left atrial pressure (LAP) and the arterial pressure (AP). From this last curve values of systolic, mean and diastolic pressures may be obtained. By processing the arterial pressure the heart rate may also be calculated, which may be corroborated with the heart rate obtained from the ECG.

4.4. Gas Exchange

4.4.a. Pulse Oximetry

Pulse Oximetry is a non-invasive method allowing the monitoring of the oxygenation of a patient's hemoglobin.

II. Physiological and Medical Background

A sensor is placed on a thin part of the patient's body, usually a fingertip or earlobe. Light with red wavelengths and light with infrared wavelengths is sequentially passed from one side to a photo detector on the other side. Changing absorbance of each of the two wavelengths is measured, allowing determination of the absorbance due to the pulsing arterial blood alone, excluding venous blood, skin, bone, muscle, fat, and fingernail polish. Based upon the ratio of changing absorbance of the red and infrared light caused by the difference in color between oxygen-bound (bright red) and oxygen-unbound (dark red or blue, in severe cases) blood hemoglobin, a measure of oxygenation (the per cent of hemoglobin molecules bound with oxygen molecules) can be made.

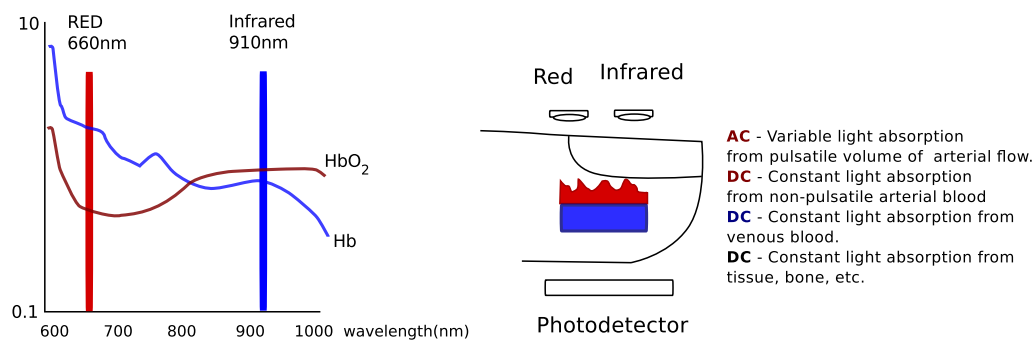


Figure II-14.: Pulse oximetry principle [2]

Oxygen Consumption

Oxygen consumption can be measured directly or calculated. Both approaches assume that all unused oxygen passes from the arterial to the venous circulation. Direct measurement of VO₂ is performed by respirometry. During respirometry, the patient breathes through a chamber that receives continuous air flow. Measurement of the oxygen depleted from the chamber, as well as the carbon dioxide and water vapor produced in the chamber, is used to determine VO₂. Respirometry can be used in mechanically ventilated patients, but the accuracy of direct measurement of VO₂ diminishes at high oxygen concentrations (eg, >80 percent).

4.4.b. Blood Gas Analysis

Blood gas analysis is a procedure to measure the partial pressure of oxygen (O₂) and carbon dioxide (CO₂) gases and the pH (hydrogen ion concentration) in arterial blood. Blood gas analysis is used to diagnose and evaluate respiratory diseases and conditions that influence how effectively the lungs deliver oxygen to and eliminate carbon dioxide from the blood. The acid-base component of the test is used to diagnose and evaluate metabolic conditions that cause abnormal blood pH. In the CPB blood gas analysis may help in knowing how effective the perfusion is.

Because high concentrations of inhaled oxygen can be toxic and can damage lungs and eyes, repeated blood gas analysis is especially useful for monitoring patients on oxygen,

for example, premature infants with lung disease, so that the lowest possible inhaled oxygen concentration can be used to maintain the blood oxygen pressure at a level that supports the patient. In intubated patients under artificial ventilation, monitoring the levels of arterial carbon dioxide and oxygen allow assessment of respiratory adequacy so that the rate or depth of ventilation, the ventilator dead space, or airway pressure can be changed to preserve the patient's optimal physiologic balance.

The measurement of arterial blood pH and carbon dioxide pressure with subsequent calculation of the concentration of bicarbonate (HCO_3^-), especially in combination with analysis of serum electrolytes, aids in the diagnosis of many diseases. For example, diabetes mellitus can be associated with a condition known as diabetic acidosis. Insulin deficiency often results in the excessive production of ketoacids and lactic acid that lower extracellular fluid and blood pH. Unattended acid-base disorders are life threatening. Acidosis is associated with severe consequences, including shock and cardiac arrest, and alkalosis with mental confusion and coma.

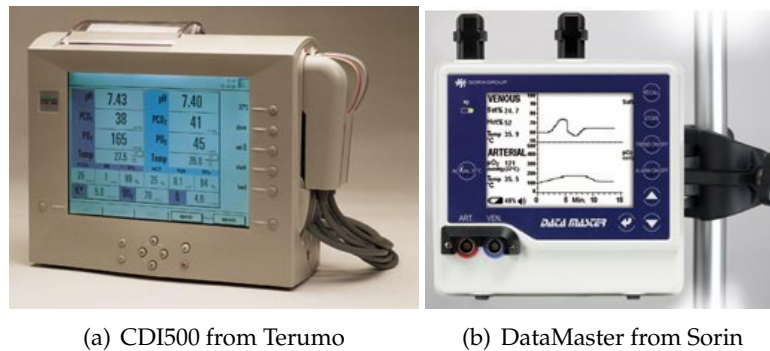


Figure II-15.: Blood gas analyzers

4.5. Temperature

Temperature may be obtained using various devices such as thermistors or infrared thermometers and in different places in the patient and system. Gas analyzers such as the CDI500 include readings of arterial and venous temperature.

4.6. Body Surface Area

The body surface area may be obtained by the Mosteller formula as:

$$BSA[m^2] = \text{sqrt}(\text{weight}[kg] \cdot \text{height}[cm]/3600) \quad (\text{II.6})$$

Using this value parameters such as the Cardiac Index may be obtained based on the following formula:

$$CI = \frac{CO}{BSA} \quad (\text{II.7})$$

4.7. Systemic Vascular Resistance

The systemic vascular resistance (SVR) refers to the resistance to blood flow offered by all of the systemic vasculature, excluding the pulmonary vasculature. This is calculated with the following formula:

$$SVR = \frac{MAP - CVP}{CO} \quad (II.8)$$

Where MAP is the mean arterial pressure, CVP is the central venous pressure and CO is the cardiac output. Since the CVP is much smaller than the MAP and sometimes close to zero this value is sometimes disregarded. The units of SVR are normally $dyn \cdot s/cm^5$; common values are between 700 and 1600 $dyn \cdot s/cm^5$. If pressure is measured in mmHg and cardiac output in L/min to obtain the SVR the result should be multiplied by 80.

4.8. Summary of Measured Signals and Normal Values

The following table shows the different signals and parameters that may be obtained from a patient and that are used to evaluate the health status of a patient.

Name	Units	Abrv.	Minimum	Maximum
Mean Arterial Pressure	<i>mmHg</i>	<i>MAP</i>	70	110
Systolic Pressure	<i>mmHg</i>	<i>SP</i>	110	140
Diastolic Pressure	<i>mmHg</i>	<i>DP</i>	65	90
Central Venous Pressure	<i>mmHg</i>	<i>CVP</i>	2	8
Cardiac Output	<i>L/min</i>	<i>CO</i>	Male 5.6	6
Cardiac Index	<i>L/min/m²</i>	<i>CI</i>	2.6	4.2
End Systolic Volume	<i>ml</i>	<i>ESD</i>	16	143
End Diastolic Volume	<i>ml</i>	<i>EDV</i>	65	240
Stroke Volume	<i>ml</i>	<i>SV</i>	55	100
Heart Rate	<i>HR</i>	<i>bpm</i>	60	220-age
Systemic Vascular Resistance	<i>dyn · s/cm⁵</i>	<i>SVR</i>	700	1600
Pulmonary Vascular Resistance	<i>dyn · s/cm⁵</i>	<i>PVR</i>	20	30
Body Surface Area	<i>m²</i>	<i>BSA</i>	Male 1.73 Female 1.6	1.9 1.6
Arterial <i>PO₂</i>	<i>mmHg</i>	<i>aPO₂</i>	92	100
Venous <i>PO₂</i>	<i>mmHg</i>	<i>vPO₂</i>	30	90
Arterial <i>PCO₂</i>	<i>mmHg</i>	<i>aPCO₂</i>	35	45
Venous <i>PCO₂</i>	<i>mmHg</i>	<i>vPCO₂</i>	10	60
Oxygen Saturation	%	<i>SpO₂</i>	92	100
Hematocrit	%	Hct	Male 42 Female 38	54 46

Table II.1.: Measurement signals with normal ranges.

Experimental and Mathematical Models

Contents

1. Experimental Model	33
2. Mathematical Model of Extracorporeal Circulation	52
3. Simulation	64

1. Experimental Model

Experiments were carried out with the objective of obtaining reference data that may be used for the analysis of the cardiovascular system undergoing ECC. These experiments give the possibility to evaluate the data acquisition devices that are used during this procedure, where they may be placed, and the type of signals that are produced. The data obtained will be used as reference data in the next section (2) to create and adapt a mathematical model of the cardiovascular system under ECC and represent what was observed experimentally.

1.1. Experimental Procedure

The experiments were approved by the Bavarian authorities, and the animals received human care in compliance with Guide for the Care and Use of Laboratory Animals (NIH publication 85-23).

1.1.a. Anesthesia and Preparation

Four domestic pigs weighing 50 ± 0.7 kg were pre-medicated with an intramuscular injection of ketamine¹ (15 mgkg^{-1}) and an atropine sulphate injection² (0.5 ugkg^{-1}). General

¹Ketanest ®, Parke Davis, Munich Germany

²Braun, Melsungen, Germany

III. Experimental and Mathematical Models

anesthesia was induced by intravenous injection of propofol³ (60 - 100 mg). Anesthesia was maintained by continuous intravenous application of propofol (10 mg/kg/h Propofol 2%) and Fentanyl⁴ (30 ug/kg/h) through a syringe pump. [15, 16]

After endotracheal intubation, the pigs were connected to a respirator and ventilated with a mixture of oxygen and air. The fraction of inspired oxygen (FIO_2) was set on 0.5.

1.1.b. Sensor placement

Figure III-12 shows the position of the different sensors used and the connection of the ECSS with the gas blender.

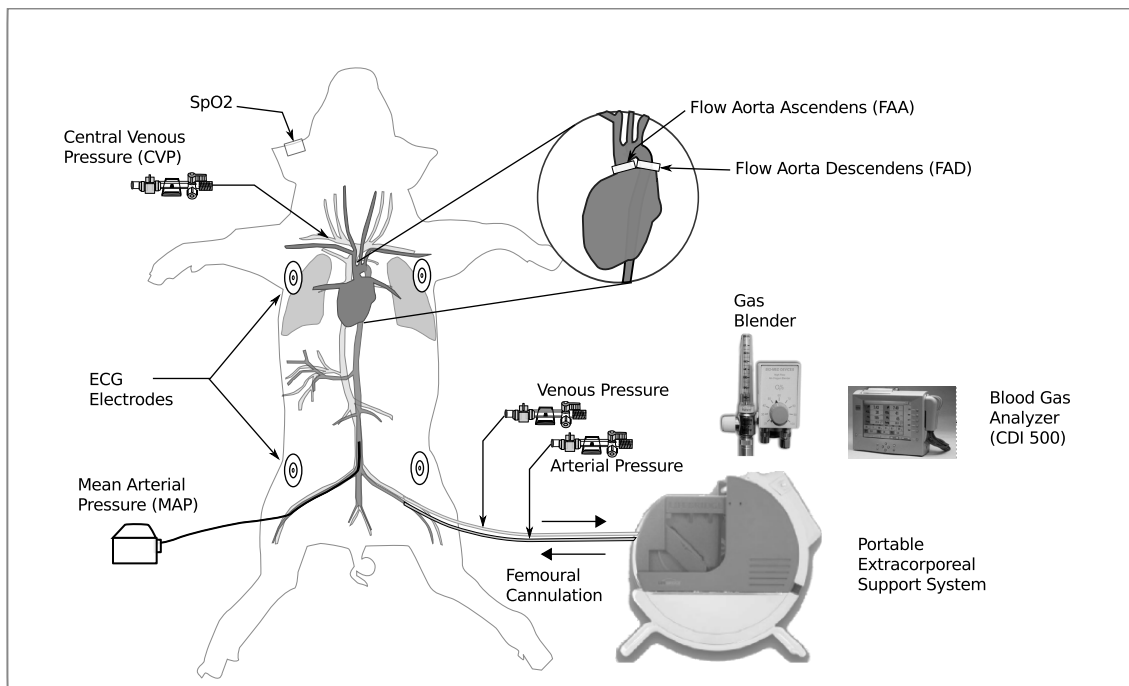


Figure III-1.: Experimental setup

A catheter was inserted into the jugular vein⁵ for monitoring of the CVP. Through the right femoral artery, a catheter tip manometer⁶ was placed in the descending aorta for monitoring the aortic pressure. Median sternotomy was done and the pericardium was opened. These pressure sensors were previously calibrated using a pressure calibration device⁷ with reference values of 0 mmHg and 100 mmHg, and were set to zero at atmospheric pressure.

To measure the aortic flow, a perivascular ultrasonic flow probe⁸ was placed at the

³Propofol, Lipuro, B.Braun Ag, Melsungen, Germany

⁴Fentanyl, Janssen Cilag, Neuss, Germany

⁵ArrowHowes™ Quad-Lumen central venous catheter, Arrow International Inc., USA

⁶Millar MIKRO-TIP® SPC350, Houston, TX, USA

⁷KAL 84, Halstrup Walcher, Kirchzarten, Germany

⁸A-Serie, Transonic Systems Inc., Ithaca, NY, USA

descending aorta above the crossing of the pulmonary veins. Another flow probe⁹ was placed in the in the ascending aorta. For heart beat detection and ECG readings a 4 lead ECG reader¹⁰ was used with 4 gel electrodes. ECG was captured at 200 Hz. An oxymeter¹¹ was placed to measure oxygen saturation and an additional reading of heart beat.

1.1.c. Cannulation and ECSS connection

Connection with the ECSS was done through femoral cannulation. From the arterial side a 20F arterial cannula¹² was introduced into the femoral artery and a 22F cannula¹³ was placed in the femoral vein.

From the ECSS additional readings of pump flow, input and output pressure and pre-oxygenator pressures were also obtained through serial communication every second.

1.1.d. Gas handling

A CDI500 gas analyzer¹⁴ was used to obtain values of Hematocrit, Hemoglobin, arterial and venous partial pressures of oxygen and carbon dioxide and arterial and venous oxygen saturation.

1.2. Experiment Protocol

The protocol that was carried out within the experiments is shown in figure III-2.

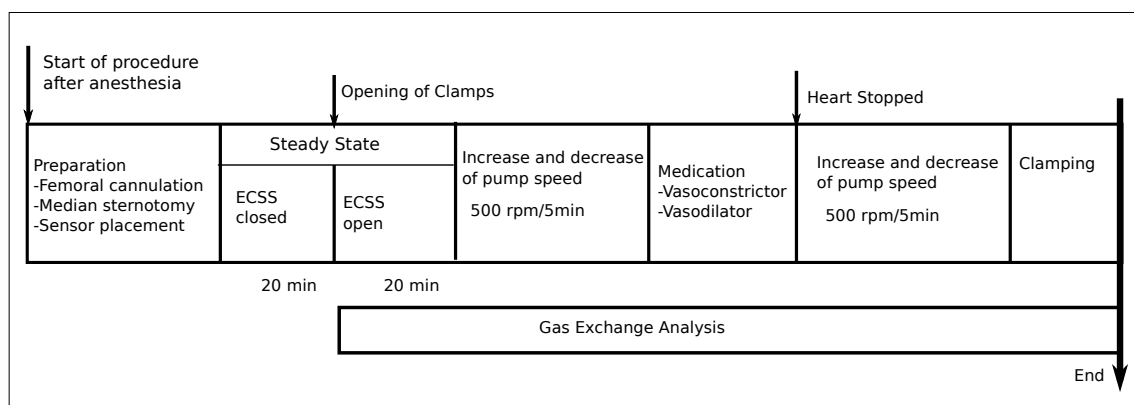


Figure III-2.: Experimental protocol

After all the sensors were in place the pig remained in a steady state (with the arterial and venous cannulas clamped) and data was acquired for 20 minutes.

The ECSS machine was previously primed with a ringer solution and the centrifugal pump was set in a continuous closed circulation at 1000 rpm.

⁹C-Serie, Transonic Systems Inc., Ithaca, NY USA

¹⁰EMI12, Corscience, Erlangen, Germany

¹¹ChipOx, Corscience, Erlangen, Germany

¹²Medtronic, Inc. Minneapolis, Minnesota, USA

¹³Edwards Lifesciences, CA, USA

¹⁴CDI500, Terumo Medical Corp, Tokio, Japan

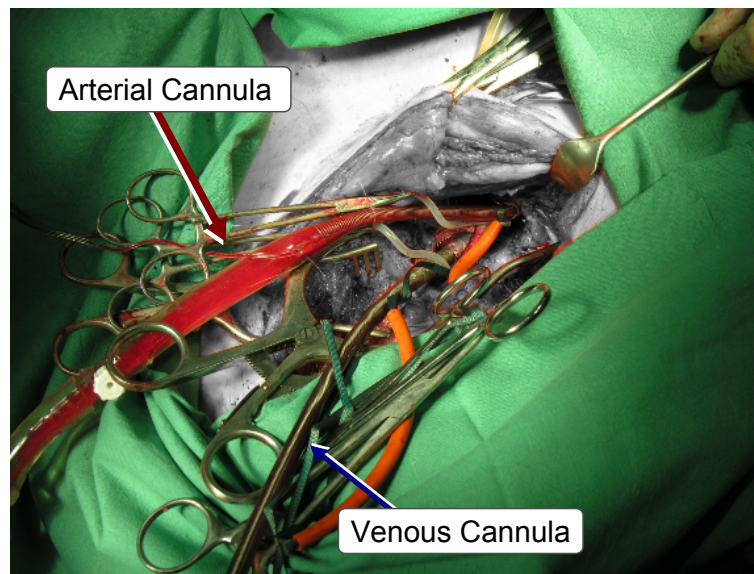


Figure III-3.: Arterial and venous cannula connected to femoral artery and vein.

The ECSS was then connected to the femoral cannulas and air was extracted from inside the tubing. Then the clamps were slowly removed and extra-corporal circulation was started increasing the centrifugal pump speed until reaching a flow of 4.5 L/m. At this point gas analysis could be started with the gas analyzer. (Figure III-3)

An additional 20 minutes of steady state with ECC was recorded at this stage. Afterwards the pump speed was changed in steps of 500 rpm with 5 minutes of recording in between.

After this state several operation tests were made to evaluate the interaction of the ECSS with the heart. The gas concentrations were analyzed at all times. Appropriate adjustments were made to the gas blender connected to the oxygenator by changing the FiO_2 and gas flow to keep the arterial oxygen partial pressure at 150 mmHg and oxygen saturation at approx 75%. The CO_2 partial pressures were kept around 45 mmHg. For the evaluation of reactions to medication a vasoconstrictor, Norepinephrine¹⁵ (NEP) (0.1 mg bolus) was administered and afterwards a vasodilator, Sodiumnirussid (SNP 10 mg bolus)¹⁶ was given. To test possible obstructions in the venous and arterial tubing both were clamped and the effects of flow and pressures were recorded.

¹⁵Arterenol, Sanovi Aventis, Frankfurt, Germany

¹⁶Niruss, UCB Pharma, Monheim, Germany

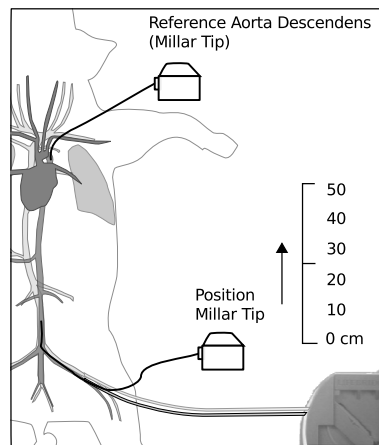


Figure III-4.: Pressure sensor positioning

1.3. Data Acquisition

All of the sensor data was recorded using software developed specifically for this experiments (AutoMedic Platform [17]). With this system it is possible to record the data that is obtained from different sensors and acquisition devices simultaneously III.5(a). Linux was used as an operating system and wxWidgets was used to create graphical user interfaces. A 16-bit data acquisition board¹⁷ was used to obtain analog data at 200 Hz. The Comedi driver [112] was used to interface with the acquisition board. A signal amplifier from BMCM¹⁸ was used to amplify the pressure transducers to be connected to the data acquisition board.

Hardware specific drivers for the ECG, SpO₂ and gas analyzer (CDI500) were developed to be directly connected to the system through serial communication.

During the experiment procedure the sensor data was visualized in a monitor application. A logging program was used to record the events that happened during the experiment procedure, adding a time stamp to each record. The data provided from other sensors were also time-stamped for post-operative analysis.

A visualization program was created to load all the captured data and display them III.5(b). This application was used to exporting specific sections of the data acquired during experimentation for further analysis.

Figure III.5(a) shows the experimental setup with the monitor and the acquisition devices.

¹⁷NI PCMCIA 6036E, National Instruments, TX,USA

¹⁸BMCM, bmc messsysteme gmbh, Germany

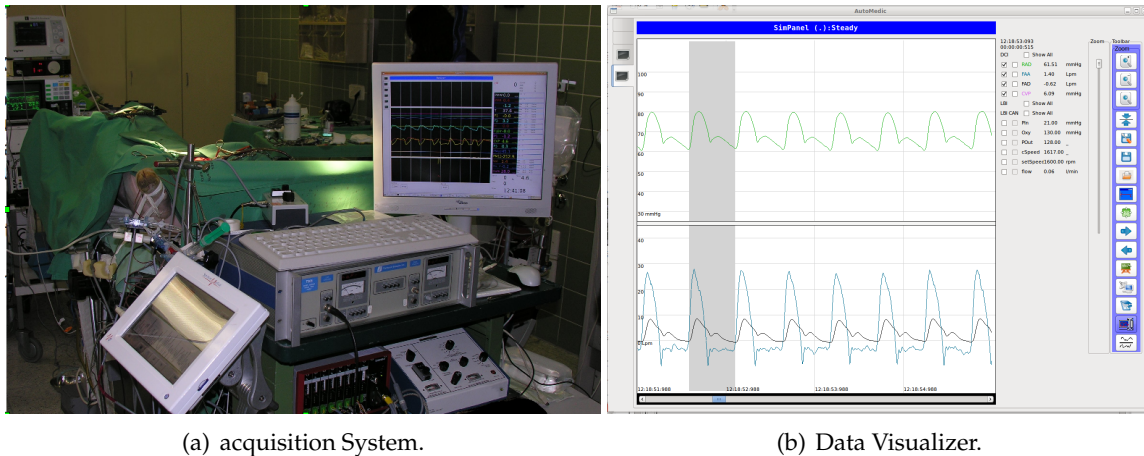


Figure III-5.: AutoMedic Platform.

1.4. Experimental Results

This section shows the data obtained from the experiment procedures considering the different parts of the protocol.

1.4.a. Steady State

This part of the procedure consisted in capturing data once all of the pressures had been placed. For 20 minutes the system remained untouched with the cannulas connected to the ECSS but the arterial and venous tubing clamped. After this the clamps were opened and the extracorporeal circulation was initiated, leaving again 20 minutes for recording of steady state.

Figure 1.4.a shows an example of the different signals obtained during steady state. The figure to the left corresponds to data captured before the ECC was initiated, with closed clamps, and the graphs to the right show the signals when the ECSS was active with an EFR of $4.3 \pm .1L/min$.

The first graph shows the data from the electrocardiogram. This signal showed some noise, possible due to the placement of the sensors, however it was possible to still detect the heart rate after some signal filtering. This heart rate could be corroborated with the heart rate detected from the pressure signal. Electrode gel also helped to improve the signal quality.

The two graphs below the ECG show the pressures obtained from the pressure transducers placed in different locations. The reference aorta descendens (RAD) shows the signal of the pressure sensor introduced through the femoral artery. (MAP) belongs to the Millar tip introduced in the arterial cannula. An additional signal was obtained directly from the arterial cannula from Medtronic. For the first case, before ECC, the arterial cannula and Millar tip are not connected. The systolic pressure was $110mmHg$ and the diastolic pressure was $80mmHg$. The central venous pressure (CVP) was $5 \pm 3mmHg$.

Once the ECSS was connected the pressures were reduced. From the RAD and MAP

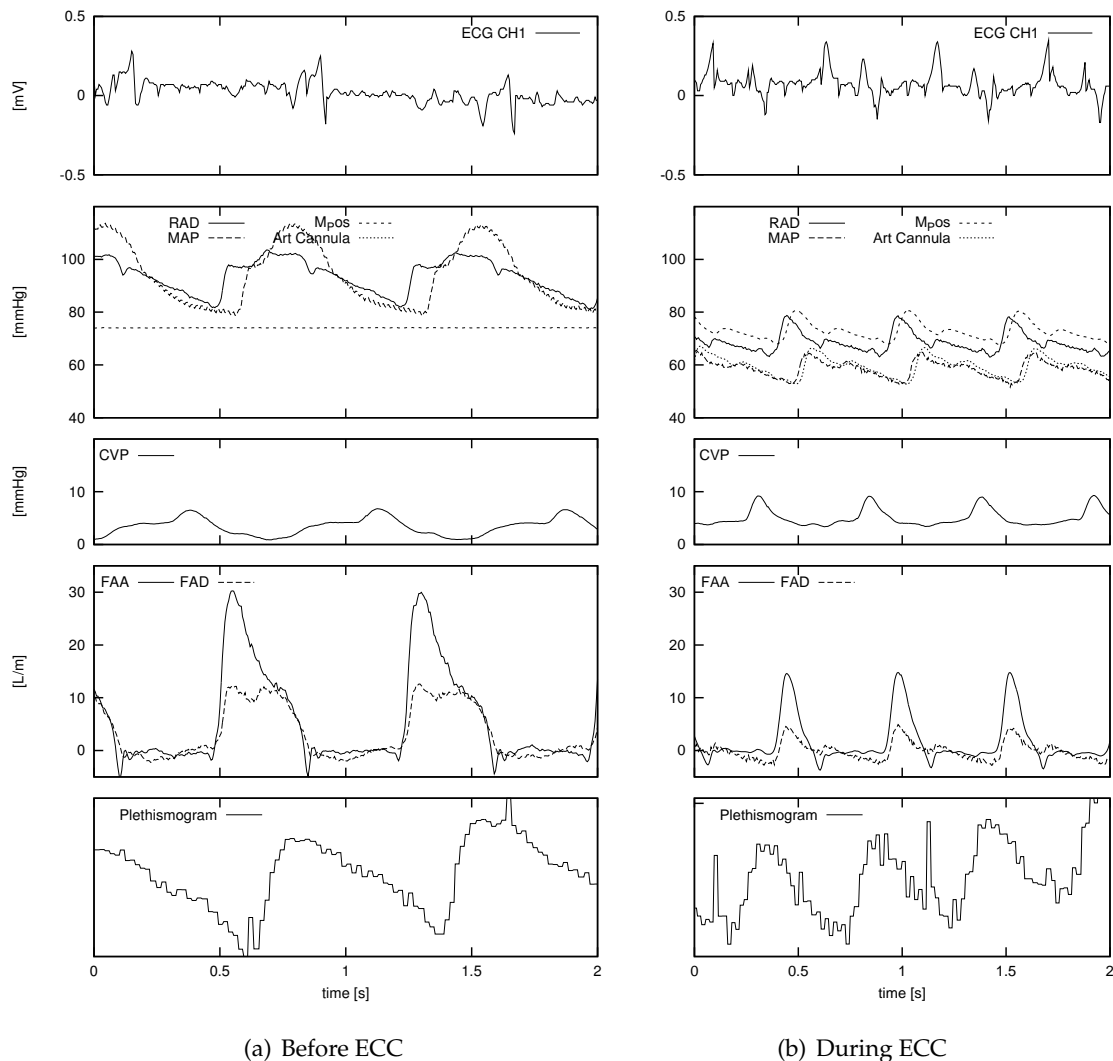


Figure III-6.: Steady states during experimentation. Before ECC(a) and after ECC(b)

sensors a systolic pressure of 80mmHg was noticed and a diastolic pressure of 65mmHg In this graph the two additional sensors are now active, showing a reduced pressure possibly due to an offset of the sensor or distance from the other sensors. The CVP was slightly increased.

The graph below the pressures correspond to the ultrasonic flow probes placed at the aorta ascendens (FAA) and aorta descendens (FAD) Before the ECC was started the maximum flow of the FAA was $30 \pm 1\text{L/m}$ with an average flow rate of $6.15 \pm .3\text{L/m}$ The flow at the aorta descendens was a maximum of $10 \pm 2\text{L/m}$ with an average of $3.5 \pm .3\text{L/m}$. During ECC the peak flow at the aorta ascendens was reduced to $15.4 \pm .4\text{L/m}$ with an average flow of $2 \pm .4\text{L/m}$ and in the aorta descendens the peak flow was at $4.9 \pm .4\text{L/m}$ with an average flow of $0.08 \pm .4\text{L/m}$. A negative peak flow of $-3.03 \pm .4\text{L/m}$ was noticed.

III. Experimental and Mathematical Models

These results and the EFR at $4.2L/min$ shows that the blood flow coming out of the heart is almost completely used to perfuse the upper body, however when the aortic valve is closed a back flow into the heart is generated caused by the constant blood flow coming from the ECSS.

The last two graphs show the plethismograph obtained from the pulse oxymeter. The sensor was placed on the ear of the pig, showing a decrease in signal strength when the ECC took place, increasing the amount of noise. However from this signal an additional heart rate may be obtained, being this sensor a non-invasive option. The SpO₂ obtained corresponded to 98% saturation. However once the heart was stopped and there was a constant flow the oxygen saturation could not be obtained any more.

1.4.b. Comparison before and after ECC

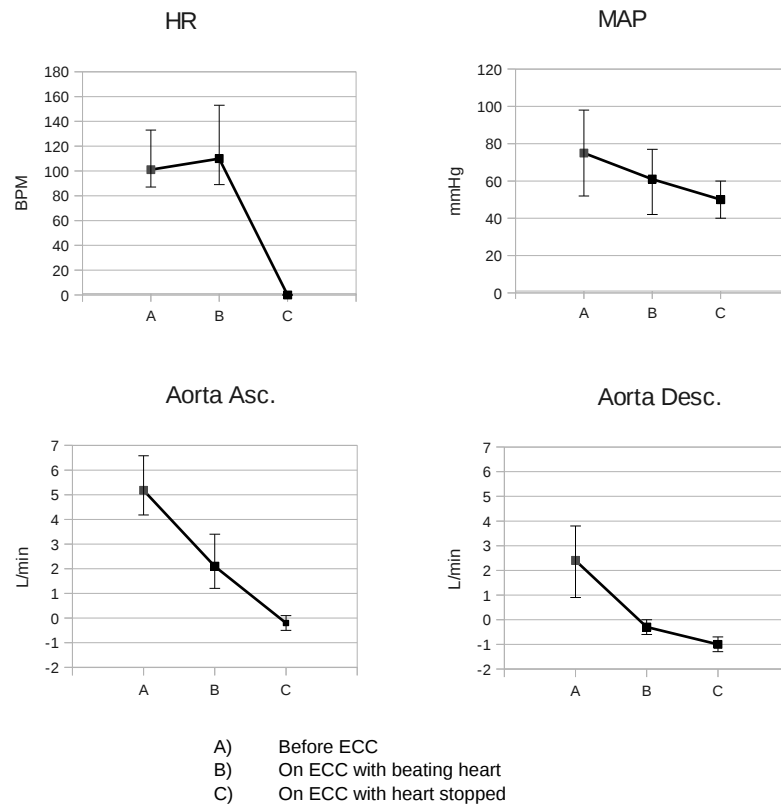


Figure III-7.: Experiment analysis of heart rate, pressure and flow before ECC, during ECC with beating heart and without beating heart.

Figure III-7 shows the tendencies for pressure, flow and heart rate for all the experiments before ECC, during ECC with a heart beating and with a heart stopped. The data was obtained as an average from all the data acquired in the different experiments.

An increase in the heart rate was noticed during ECC however the MAP decreased. The reason for this could be a reduction of viscosity of the blood due to a mix with the priming solution, another reason for pressure decrease may be due to the invasiveness of the operation procedure and loss of blood. The cardiac output from the heart also decreased once the ECSS was connected. This may be visualized from the flow in the aorta ascendens. When the heart was fully stopped the aorta ascendens showed a flow close to $0L/m$. A slight negative flow could be appreciated, indicating an backflow into the heart. The sensor at the aorta descendens showed that part of the EFR (approx. $1 \pm 0.5L/min$) was used to perfuse the upper part of the body. This amount of blood was less than when the heart was active.

III. Experimental and Mathematical Models

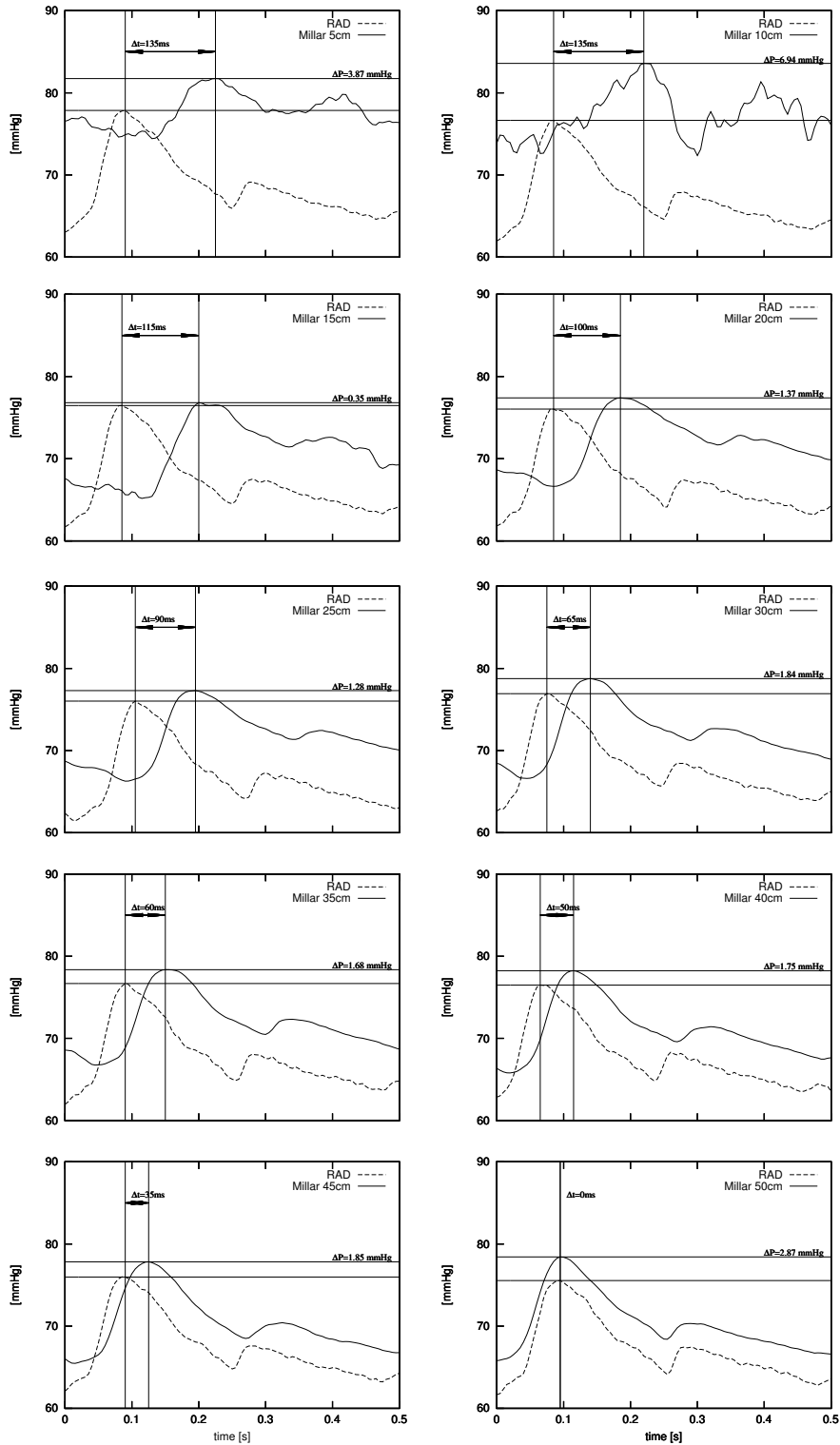


Figure III-8.: Results pressure sensor positioning.

1.4.c. Positioning

This step in the protocol was used to evaluate the differences between placing a pressure sensor at the beginning of the cannula to the pressure that is obtained closer to the heart. The positioning of the sensor was described in figure III-4. The graphs on figure III-8 shows the pressure at the aorta descendens and a millar tip sensor introduced through the arterial cannula.

The top graph to the left shows the sensor at 5 cm from the inlet of the cannula. In every step 5 centimeters were introduced until reaching 50 centimeters, shown at the bottom right of the figure. From the first two graphs we can see the pressure in the cannula has no relation to the arterial pressure and some noise was added due to the blood flow causing turbulence and movement of the sensor. Once the sensor is out of the cannula and into the artery (at 15 centimeters) the shape of the curve becomes similar for both sensors. What may be noticed is that there is not a direct relation between the difference of the minimum and maximum values of the pressure curves and the positioning of the sensor, these values remain almost constant. However a shift of phase in the pressure curve may be observed in relation to the position of the sensor. The closer the millar tip is to the pressure sensor at the aorta descendens less phase offset exists between both curves. When the tip is introduced 50 centimeters we may assume both sensors are at the same position and the phase shift disappears. A value offset between the sensors is observed of $1.7 \pm .5mmHg$ possibly due to calibration.

With these results we may conclude that a sensor placed at the tip of the cannula is sufficient to have knowledge of the pressure curve, and one should just consider the phase offset to the arterial pressure at the outlet of the heart if it is important to correlate it with other curves such as ECG or blood flow.

1.4.d. Different Pump Speeds

By gradually changing the pump speed of the ECSS it is possible to analyze the relation between pressure, flow and pump speed. Figure III-9 shows the location of the pressure sensors in the machine. One is before the reservoir (P In), a second is between the centrifugal pump and the oxygenator (Pre-Oxy) and a third is after the arterial filter (P Out). Additional pressure sensors are placed at the inputs of the arterial cannula (Art Can) and venous cannula (Ven Can). The flow sensor is shown as EFR also after the arterial filter.

Figure III-10 shows the results of two different experiments when the centrifugal pump speed was increased gradually starting from 1500 rpm and up to 3500 rpm. The top graph shows the inlet, outlet and pre-oxygenator pressures of the ECSS and the pressures at the arterial and venous cannula. We can see that the highest pressure is before the oxygenator since it is right after the centrifugal pump. The pressure drop between this pressure and the outlet is due to the resistance caused by the oxygenator and the arterial filter. From the venous side there is not a big difference between the venous cannula and the inlet of the ECSS, in between these two sensors there is only tubing and the reservoir.

The graph in the middle shows the mean arterial pressure. For the first experiment MAP started at $62 \pm 4mmHg$ and was increased to $70 \pm 4mmHg$ almost with a linear behavior. The graph from experiment 2 shows a higher starting MAP, and was not as affected by the increase of pump speed.

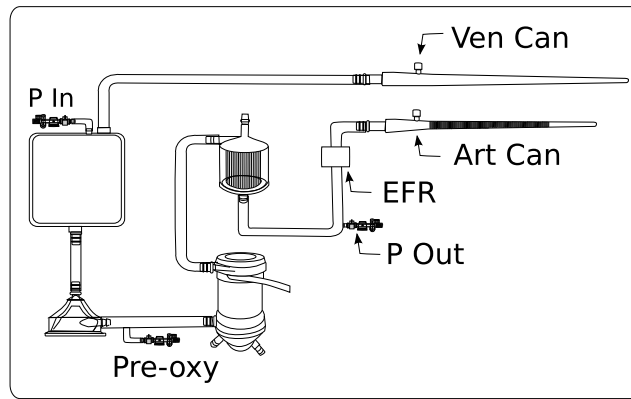


Figure III-9.: Pressure sensors

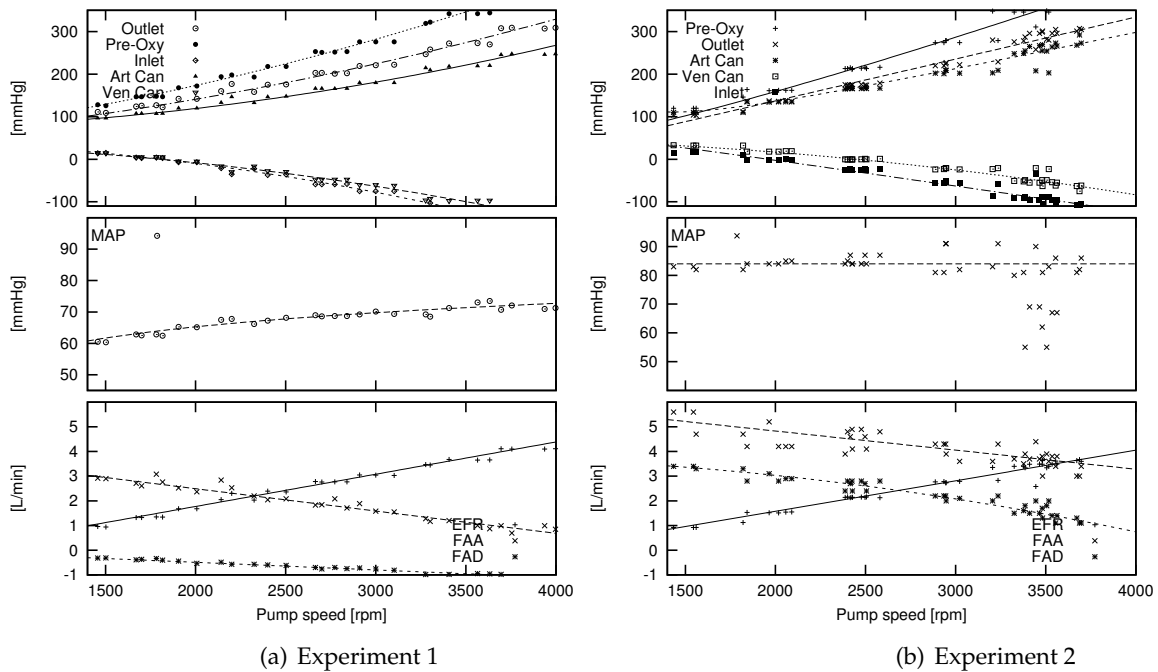
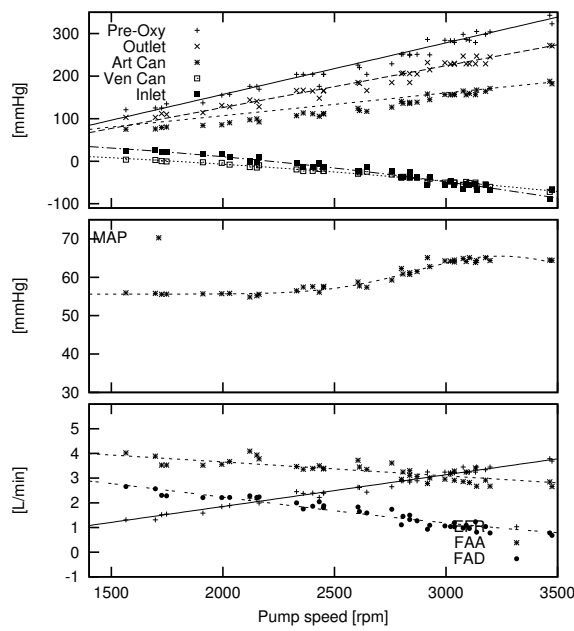
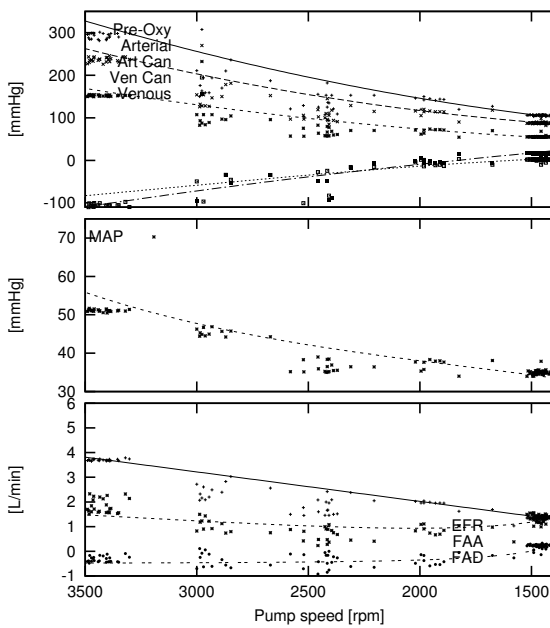


Figure III-10.: Change of centrifugal pump speeds with active heart in different experiments.

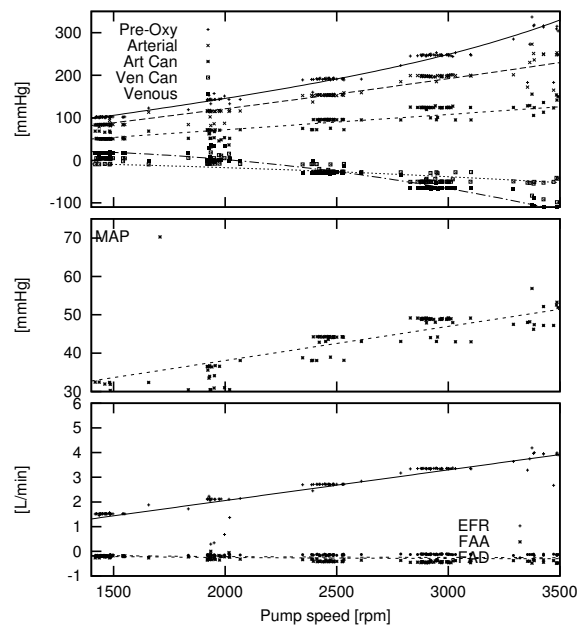
The bottom graphs shows the data from the flow sensors. Both cases show a linear increase of EFR reaching a flow of $4 \pm 0.5 L/min$ at full pump speed. For the first case the flow at the aorta descendens started at $3 \pm 0.2 L/min$ and was reduced to approx. $1 L/min$. The flow at the aorta descendens was already negative at low speed. Experiment 2 started with a much higher flow at the aorta ascendens at $5.4 \pm 0.4 L/min$ and was reduced to $3.8 \pm 0.4 L/min$ with full pump speed.



(a) Heart active starting ECC



(b) Heart active after 1.5 hours of ECC



(c) Heart stopped

Figure III-11.: Change of centrifugal pump speed in experiment 3 at different stages of the protocol.

Figure III-11 shows the results of changing pump speed in a single experiment during different parts of the procedure. The top graphs show the pressure and flow curves exactly when the ECC was started and the clamps were opened. At this point the cardiac output was of $4 \pm .4 L/m$. With the pump speed at full speed and EFR at $4 L/m$ the cardiac output

III. Experimental and Mathematical Models

was reduced to $3L/min$. After the procedure had run for 1 hour and 30 minutes the pump speed was again changed. At this point however the MAP was at $52mmHg$ and the cardiac output was reduced to $2L/min$ the flow at the aorta descendens was already negative showing a weakening of the heart. When the pump speed was reduced to 1500 rpm the MAP was reduced to $38 \pm 5mmHg$. Four hours after ECC the heart was stopped. At this point the pump speed was again changed. The MAP was as low as $32 \pm 4mmHg$, at full speed this pressure was increased to $45 \pm 10 mmHg$. Flow of the aorta ascendens was close to $0L/min$. The aorta descendens however also was not affected.

1.4.e. Gas Analyzer

The gas analyzer was used to change FiO_2 and gas flow parameters in the blender. The values obtained in the gas analyzer were in general stable, however sudden changes in flow also affected the gas partial pressures.

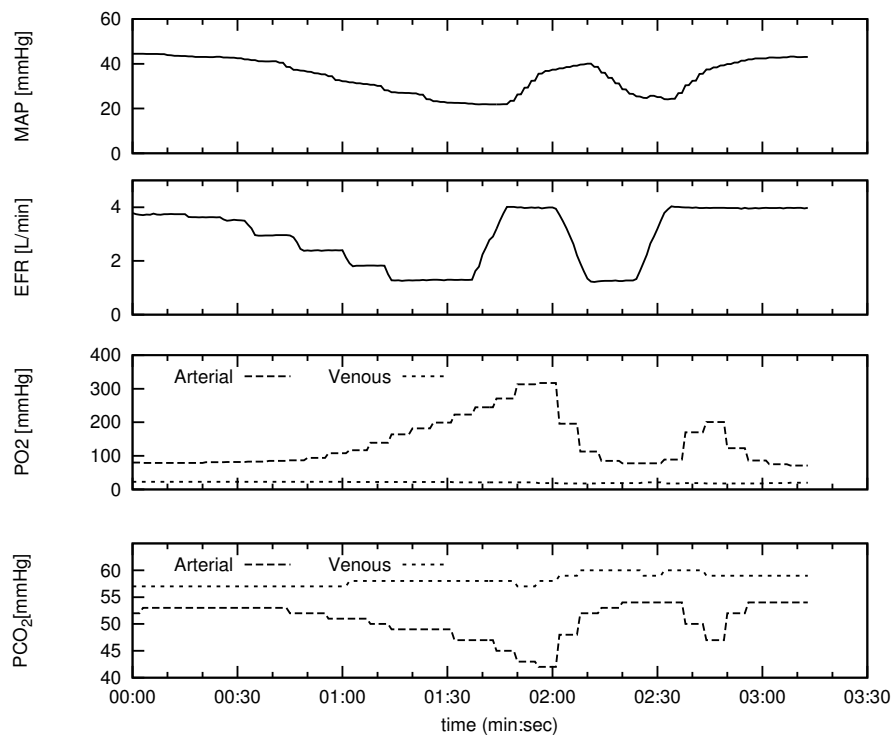


Figure III-12.: Oxygen and Carbon Dioxide partial pressures in relation to MAP and EFR

Figure III-12 shows the O_2 and CO_2 partial pressures obtained from the gas analyzer. In this case the EFR was reduced from approx. $4 L/min$ to $1.5L/m$. The FiO_2 and gas flow remained constant at 60% and 6 L/min respectively. This produced a gradual increase of the O_2 partial pressure in the arterial line until reaching $300 \pm 20mmHg$. The venous PO_2 remained constant. Once the EFR was restored the PO_2 at the arterial side was reduced back to approximately $100mmHg$. The bottom graph presenting the PCO_2 shows a value of $57 \pm 5mmHg$ at the venous side and $53 \pm 5mmHg$ at the arterial side. When the EFR

was reduced the arterial PCO_2 decreased to $43 \pm 5 \text{ mmHg}$ given the constant gas exchange in the oxygenator. The venous PCO_2 slightly increased. Once the EFR was restored the arterial PCO_2 returned to its original value of approximately 53 mmHg. An additional reduction of EFR was done afterwards. From this a delay of 10 seconds was noticed between the signal of the EFR being again increased and the actual effect in the gas partial pressures.

1.4.f. Medication

The following results show the effects of introducing a vasoconstrictor drug for several experiments and a case of using a vasodilator drug.

Vasoconstrictor

Figures III-13 and III-14 show the effects of introducing Norepinephrine as a vasoconstrictor drug in different experiments.

For the calculation of the SVR the EFR was considered together with the arterial flow FAA. The heart rate was extracted from the arterial pressure curve.

Figures III.13(a), III.13(b) and III.14(a) show the reaction when 0.1mg of NEP were added.

In all the cases the heart rate was increased above 200 bpm and then slowly decreased until reaching a stable heart rate. The mean arterial pressure was also increased, however the maximum pressure obtained was not the same for all cases. In the first case the MAP

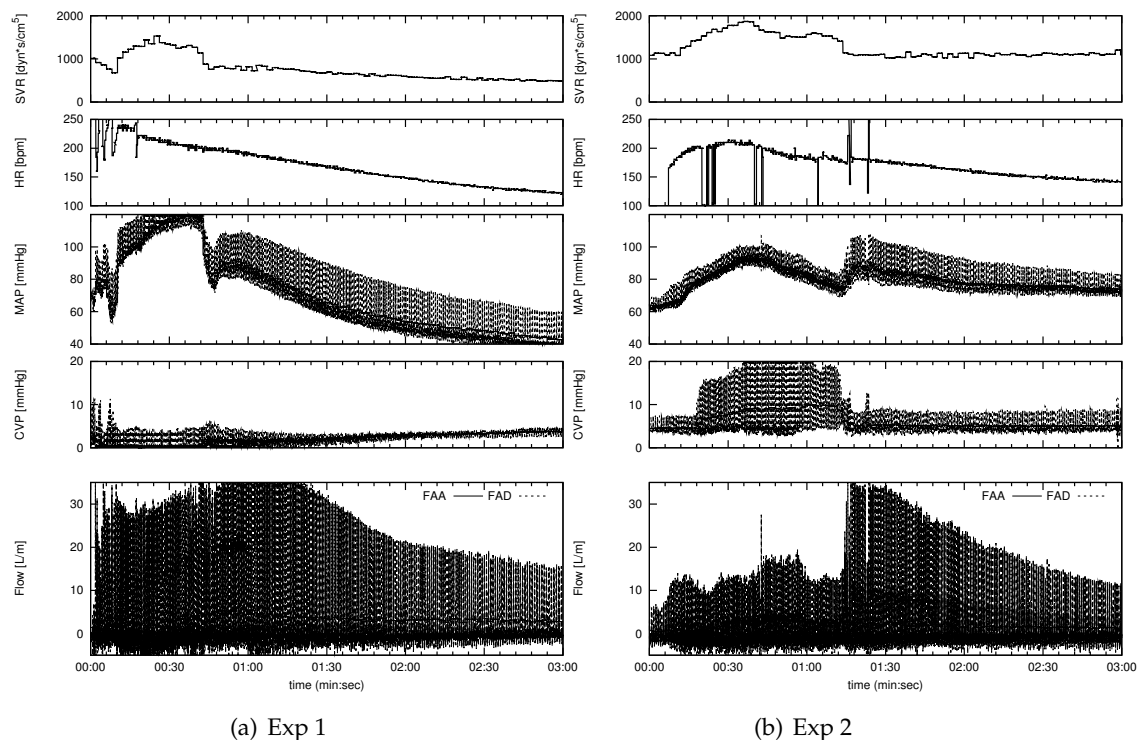


Figure III-13.: Response to vasoconstrictor for different experiments.

III. Experimental and Mathematical Models

reached a maximum of $120 \pm 5 \text{ mmHg}$. An increase of $30 \pm 20 \text{ mmHg}$ was noticed from the previous value of MAP. For the first two experiments a sudden decrease in pressure may be noticed, this was due to a sudden change in pressures from the ECSS. In experiment two we can notice before this sudden change the CVP had increased, showing peaks of 20 mmHg . This may also happen when the cannula is collapsing due to a very negative pressure. After pressure was restored the CVP decreased. This also produced an increase of cardiac output and MAP.

Figure III.14(b) shows the reaction when 0.2 mg of NEP were added after the first dosage of 0.1 mg . An increase of pressure was noticed, from a peak of $80 \pm 5 \text{ mmHg}$ to $85 \pm 5 \text{ mmHg}$. This shows that the effect is not directly doubled. Additional experiments show that after giving the same amount of dosage twice with an interval of 5 minutes, similar effects may be obtained.

Further studies may evaluate the effect of giving a continuous dosage instead of a single bolus.

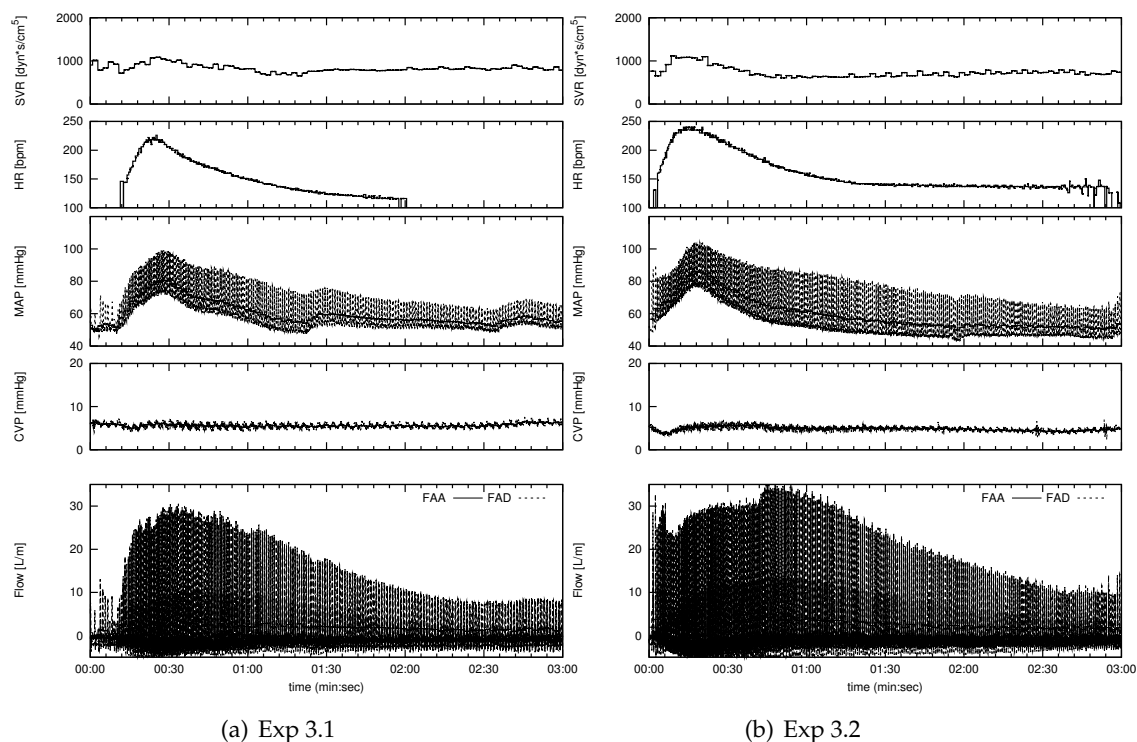


Figure III-14.: Response to vasoconstrictor in single experiment.

Vasodilator

Figure III-15 show the experiment results of using SNP as a vasodilator drug. The heart rate did not show any significant change during this procedure. There was a decrease in MAP from $65 \pm 4\text{mmHg}$ to $55 \pm 4\text{mmHg}$, however this effect was not noticed in the SVR since the cardiac output was already low. In this particular case the heart rate was already weak and after 2 minutes the pump speed was increased, changing the effect of the drug, however this shows the possible counter effect of using a drug as a vasodilator, without affecting the heart rate.

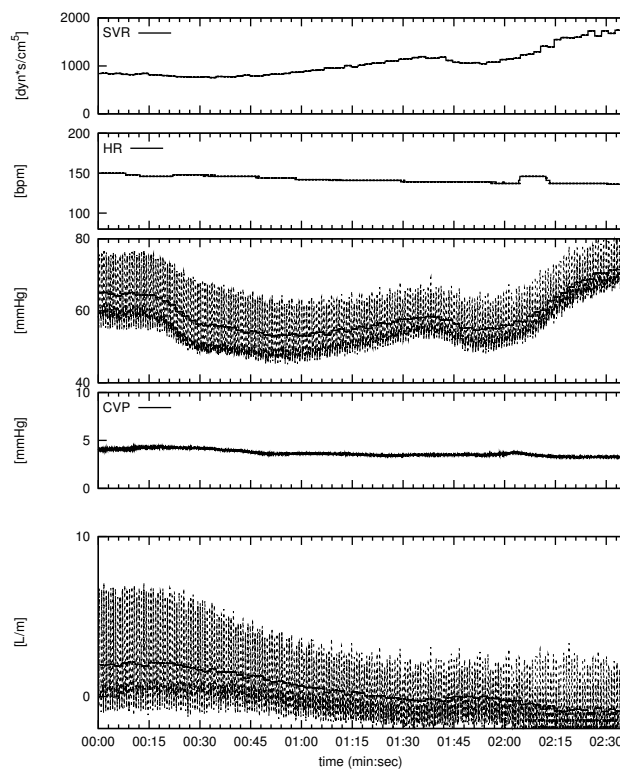


Figure III-15.: Vasodilator

1.4.g. Cannula Obstruction

This section shows the results of sudden obstruction of the arterial and venous cannula. This analysis is important since the system as a portable device will be moved and arterial or venous tubing may be obstructed. This should be quickly identified to take immediate action and avoid complications.

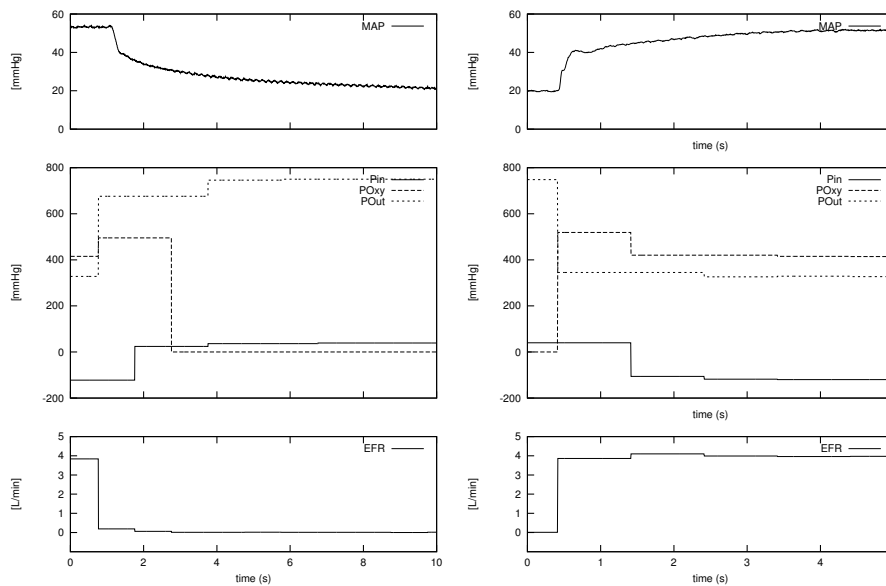


Figure III-16.: Obstruction arterial cannula. (Left = Closing / Right = Opening)

Figure III-16 shows the results for the case of closing the arterial cannula. This made an immediate decrease in MAP and EFR. The sensor of the output pressure registered an increase up to almost 800 mmHg. This should have also been noticed in the pressure after the oxygenator, however the sensor was on saturation showing 0 mmHg. After the arterial cannula was open again the flow was restored to 4 L/min and the MAP jumped to 40 mmHg and slowly increased to $50 \pm 4 \text{ mmHg}$ and the pressure sensor before the oxygenator was again higher than the output pressure.

Figure III-17 shows the effect of obstructing the venous cannula. We can see a contrary effect than the arterial obstruction in the MAP by having a sudden increase. The reason for this is that the blood is being still pumped into the body, but it is not drained out, bringing back the blood into the patient. The blood that is still pumped into the body is what is in the reservoir. The flow was gradually decreased at this point until the reservoir is empty. After the venous tubing was opened the centrifugal pump was not able to generate any more flow since it was already full of air. This type of obstruction presents more danger than the arterial obstruction since the system may not be restored after the tubing has been reopened.

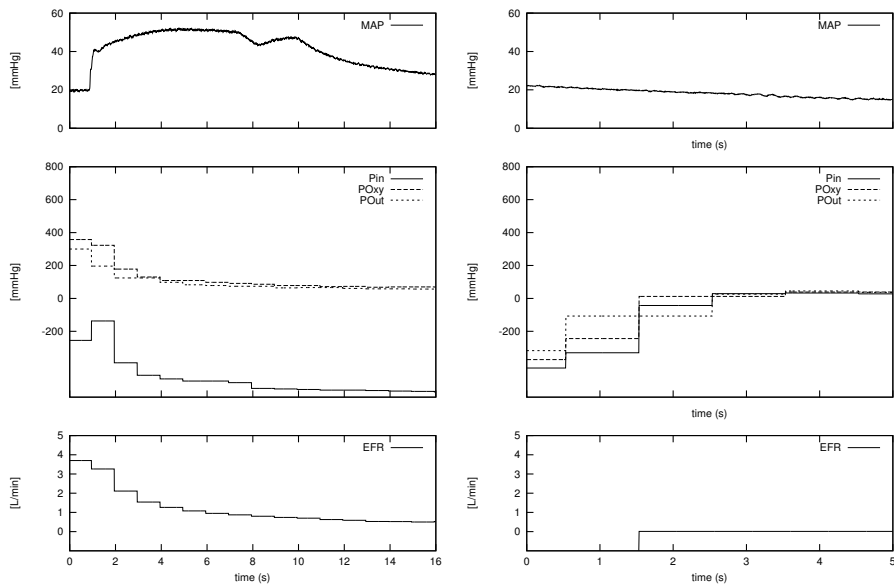


Figure III-17.: Obstruction venous cannula.(Left = Closing / Right = Opening)

1.5. Remarks

From the analysis of the different experiments it was noticed that when the heart is stopped and the ECSS takes full charge of perfusing the body the flow going to the upper body is significantly reduced together with the MAP. These parameters however could not be increased in the previous experiments since the ECSS was already running at full speed. The analysis done in the next section of modeling of the ECSS will show that one of the biggest resistances in the system causing a high output pressure and low flow is the cannulation.

2. Mathematical Model of Extracorporeal Circulation

This section contains the designed mathematical model used to describe the cardiovascular system together with an ECSS to resemble extracorporeal circulation. The experimental data from the previous section was used to adjust the model parameters. The purpose of this model is to integrate this model into a simulation environment that is capable of interacting with a control system.

2.1. Previous Cardiovascular and ECC Mathematical Models

Extensive research has been previously done in ways to represent in a mathematical manner the human physiology [38–48, 48–51, 94]. These models may help to understand the mechanics and functions of the cardiovascular and pulmonary systems, giving the possibility to provide better diagnosis and possible treatments. Since 1966 McLeod[38] created Physbe as a physiological simulator. However with the processing capabilities of computers at that time it was difficult to create sophisticated systems. With the advances in computing and further development of these models different projects were developed such as the Physiome project [39] focused on describing the physiological dynamics of the human body. These models however may be highly complex and do not consider extracorporeal circulation. Further developments by Bauernschmitt described the mathematical modeling of extracorporeal circulation[53] which served as ground work for this work. Lu et al[54] created a human cardiopulmonary system which includes a baroreceptor model and airway mechanics together with gas exchange model. Brawany[61] described a cardiovascular model with malfunctions and included the effect of different drugs. Luo et al[46] created a cardiopulmonary to study and predict normal and diseased ventricular mechanics.

With the knowledge of the previous work mentioned a new model was created. The considerations for the construction of this model were to include the main physiological interactions that are involved in ECC while still keeping the model simple to be able to adjust a reduced number of model parameters to represent the experimental data.

2.2. Mathematical Model Overview

The mathematical model was comprised of several subsystems. The main component is a model of the cardiovascular system connected to an ECSS. This calculates the haemodynamics and blood distribution. A gas exchange model consisting of six compartments is used for the calculation of oxygen delivery and CO_2 extraction. A baroreceptor model was added to regulate the heart rate, and a medication model for possible vasoactive drugs delivery (Figure III-18). Possible illnesses were introduced in the model by modifying specific parameters of the different systems.

2.3. Cardiovascular Model

A simplified cardiovascular system was modeled characterized as a series of hydraulic compliant elements (C) interconnected by resistance elements (R). The heart was com-

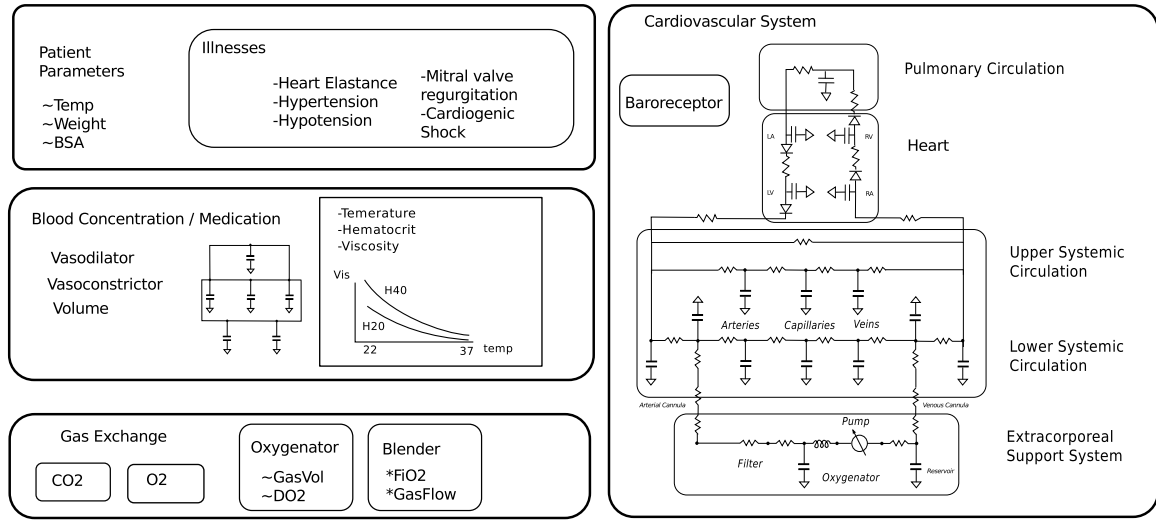


Figure III-18.: Mathematical model. Subsystem diagram.

posed of a four compartment element with the corresponding valves represented as diodes [43–51]. The relation of pressure (P), volume (V) and flow (Q) was used in relation to the neighbor component considering resistance (R), compliance (C) and inductance (L) elements. These relations were expressed in the following formulas:

$$P_i = (V_i - V_{i-1}) \cdot 1/C_i \quad (\text{III.1})$$

$$Q_i = (P_i - P_{i-1})/R_i \quad (\text{III.2})$$

$$dV_i/dt = Q_{i-1} - Q_i \quad (\text{III.3})$$

$$dQ_i/dt = (P_i - P_{i-1}) \cdot 1/L_i \quad (\text{III.4})$$

The pulmonary circulation consists of a single compliance and resistance. The systemic circulation was separated in two sections; the first corresponding to the upper body, including the cerebral circulation and arms, and the second corresponding to the lower body including legs torso and rest of the body. For each of these two systemic sections three compartment elements were used to represent the arteries, capillaries and veins (Figure III-19).

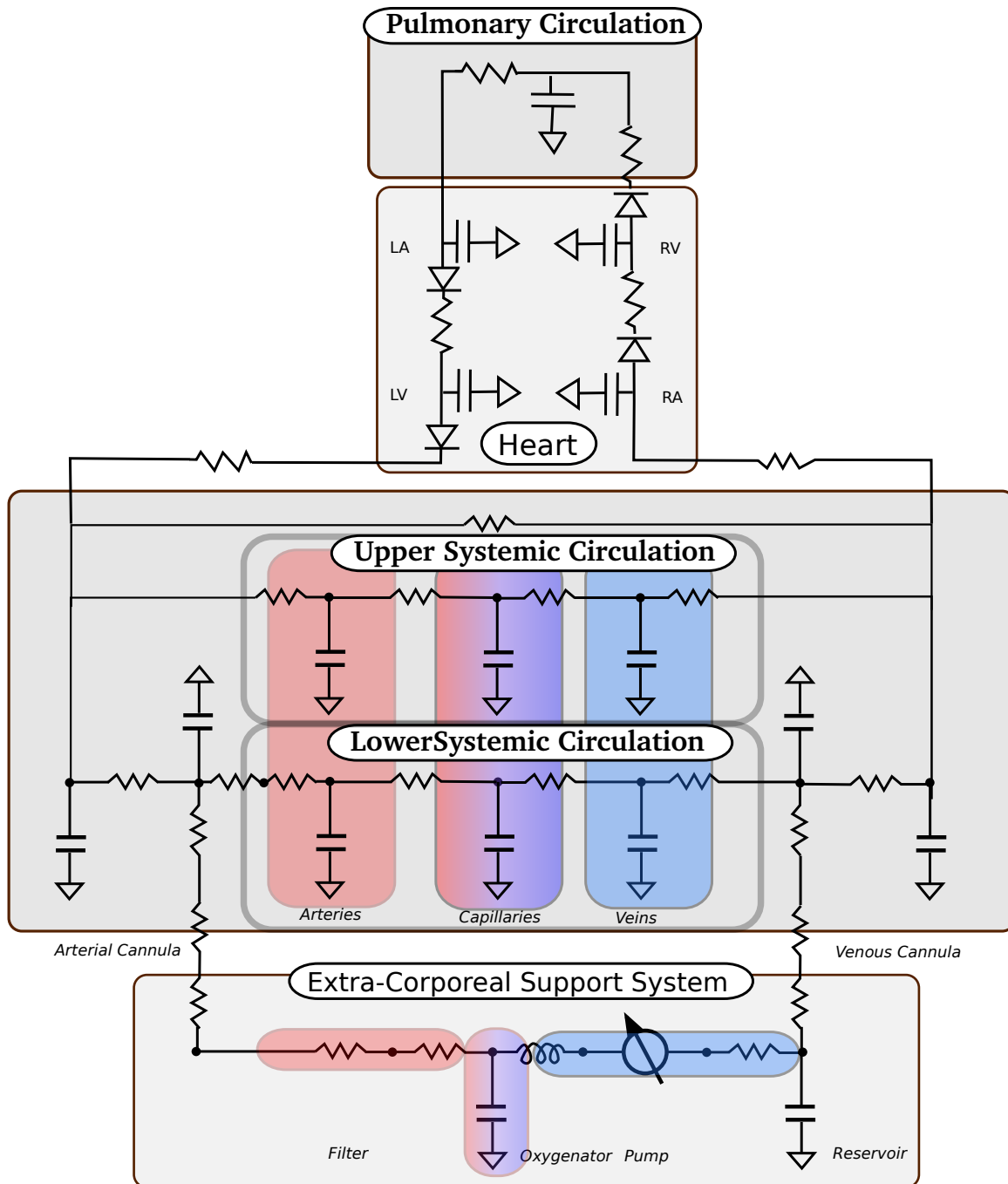


Figure III-19.: Complete Cardiovascular Model

2.3.a. Heart Model

The ventricles of the heart are modeled as a three-walled system, the right ventricular (RV) and left ventricular (LV) free walls and the coupling septal wall. This creates four functional volumes.

The heart contractions were modeled by changing the elastance of each compartment and a sinusoid activation signal $y(t)$ with a frequency provided by the baroreceptors to established the heart rate (III.5).

$$E(t)_i = E_{iMax} - E_{iMin}x(y(t)_i + E_{iMin}) \quad (III.5)$$

2.3.b. Baroreceptor Model

Four functional blocks are used: the baroreceptor, the central nervous system (CNS), the efferent pathways and the effector organ. The input to the baroreceptor element is the mean arterial pressure and the output is the instantaneous firing frequency of the baroreceptor. This is represented by a sigmoidal baroreflex curve (bfc) function as presented in equation III.6 as described by Ursino et al. and Brawany [60, 61].

$$bfc(t) = \frac{e^{\gamma(MAP-MAP_0)}}{1 + e^{\gamma(MAP-MAP_0)}} \quad (III.6)$$

MAP_0 is the nominal value of MAP; γ is an empirical constant ($0.06263 \text{ mmHg}^{-1}$).

The CNS is the main physiological controller of the body. Its input is the firing rate and produces the efferent sympathetic and parasympathetic nerve activities. The sympathetic tones modify the heart contractility (E_{max}) and the heart rate is modified in response to both the sympathetic and vagal activities. The heart period changes induced by sympathetic stimulation affected by a low-pass first order dynamics shown in equations III.7 to III.9.

$$\frac{d\sigma}{dt} = \frac{HR}{180} \cdot \frac{(bfc - \sigma)}{4} \quad (III.7)$$

$$\frac{d\Delta Hr}{dt} = \frac{1}{\tau_\theta} \cdot (-\Delta Hr(t) + \sigma(t)) \quad (III.8)$$

$$HR(t) = \Delta Hr(t) + Hr_0 \quad (III.9)$$

σ determines a delay effect over bfc; this value is used for the calculation of the increase/decrease of heart rate over time (ΔHr). ΔHr is then added to a heart rate base value (Hr_0).

2.4. Gas Exchange Model

For the calculation of gas exchange the model was further reduced to a six compartment system: a systemic artery, capillary and vein compartments, a pulmonary compartment, a reservoir compartment and a compartment for the oxygenator in the ECSS (figure III-20).

Oxygen is carried by the blood in two forms: most carried combined with hemoglobin and a smaller amount dissolved in the plasma. Each gram of hemoglobin may carry 1.34

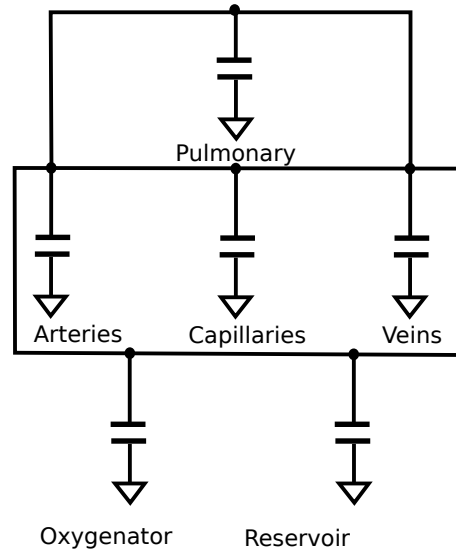


Figure III-20.: Six compartment model used for gas exchange and medication model.

ml of oxygen when fully saturated. To calculate the actual oxygen concentration in the blood for each compartment the following formula was used:

$$CtO_{2,i} = Hb \times 1.34 \times SpO_2 + PO_{2,i} \times 0.003 \quad (III.10)$$

CtO_2 is measured in mlO_2/dl , Hb is the hemoglobin concentration measured in g/dl ; 1.34 is the oxygen binding capacity of blood with units $mlO_2/gmHb$, SpO_2 is the oxygen saturation, PO_2 is the oxygen partial pressure in $mmHg$ and 0.003 is the solubility of oxygen in the blood in $mlO_2/mmHg/dl$.

The relation between oxygen partial pressure and oxygen saturation was approximated using the Hill coefficient nH in the following formula:

$$SpO_2 = \frac{(PO_2/P50)^{nH}}{(PO_2/P50)^{nH} + 1} \quad (III.11)$$

The change of concentrations in each compartment for oxygen and carbon dioxide were calculated based on the following formula:

$$\frac{dCt_{G,i}}{dt} = \frac{\dot{V}_G \cdot (CtO_{2,i-1} - CtO_{2,i})}{V_i} \quad (III.12)$$

2.4.a. Oxygenator

The gas exchange in the oxygenator was modeled considering the partial pressures of the blood compartment with an additional gas compartment that is connected to an air-oxygen blender.

$$\frac{dP_G}{dt} = \frac{\dot{V}_G \cdot (P_{G,i} - P_G) + P_{atm} \cdot D_G \cdot (BloodP_{G,i} - P_G)}{V_G} \quad (III.13)$$

The subindex G corresponds to either O_2 or CO_2 . \dot{V}_G is the flow of the gas mixture introduced into the oxygenator and is determined by the oxygen blender. $P_{G,i}$ is the input partial pressure of O_2 (PO_2) and CO_2 (PCO_2). The input of PCO_2 is 0 mmHg. The input PO_2 is determined by the oxygen blender obtained by multiplying FiO_2 times the atmospheric pressure P_{atm} (760mmHg). D_G is the diffusion factor for O_2 and CO_2 . $BloodP_{G,i}$ corresponds to the gas partial pressure of the blood entering the oxygenator. This is obtained from the gas partial pressures at the pulmonary artery from the gas exchange model of the cardiovascular system.

The input oxygen partial pressure is based on the configuration of the FiO_2 in the blender and is calculated with the following formula:

$$PO_{2in} = FiO_2 \cdot PO_{2ref} \quad (III.14)$$

Where PO_{2ref} refers to the atmospheric pressure. (760mmHg at sea level).

2.5. Extracorporeal Circulatory Support System

From the lower systemic circulation before the arterial compartment and after the venous compartment the ECSS was connected by introducing additional resistances representing the femoral artery and vein. From this points the arterial and venous cannulas were attached respectively.

As mentioned in section 2.3 of chapter II the main components of an ECSS consists of a reservoir, a centrifugal pump, an oxygenator and an arterial filter. Two compartment elements and two resistances were used to represent the reservoir and the oxygenator. Since the Lifebridge system [23] was used for ECC during our experiments it was taken as a reference to construct the model of the ECSS. To achieve this first all of the components of the machine were taken apart and analyzed separately. Figure III-21 shows an example of the experimental setup for the filter evaluation. The experimental data obtained from these procedure is shown in figure III-22. A quadratic function was used to model the response of flow against pressure as shown in equation (III.15).

$$\Delta P_i = K a_i Q^2 + K b_i Q \quad (III.15)$$

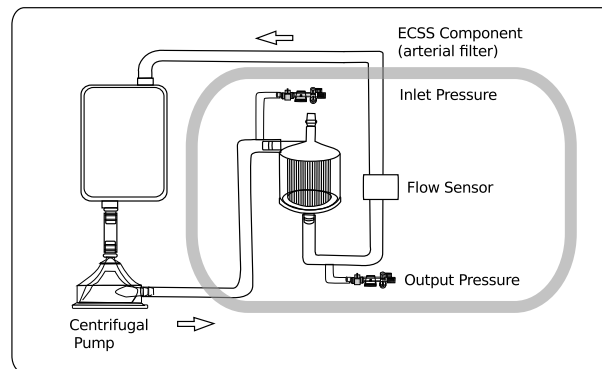


Figure III-21.: Experimental Setup

III. Experimental and Mathematical Models

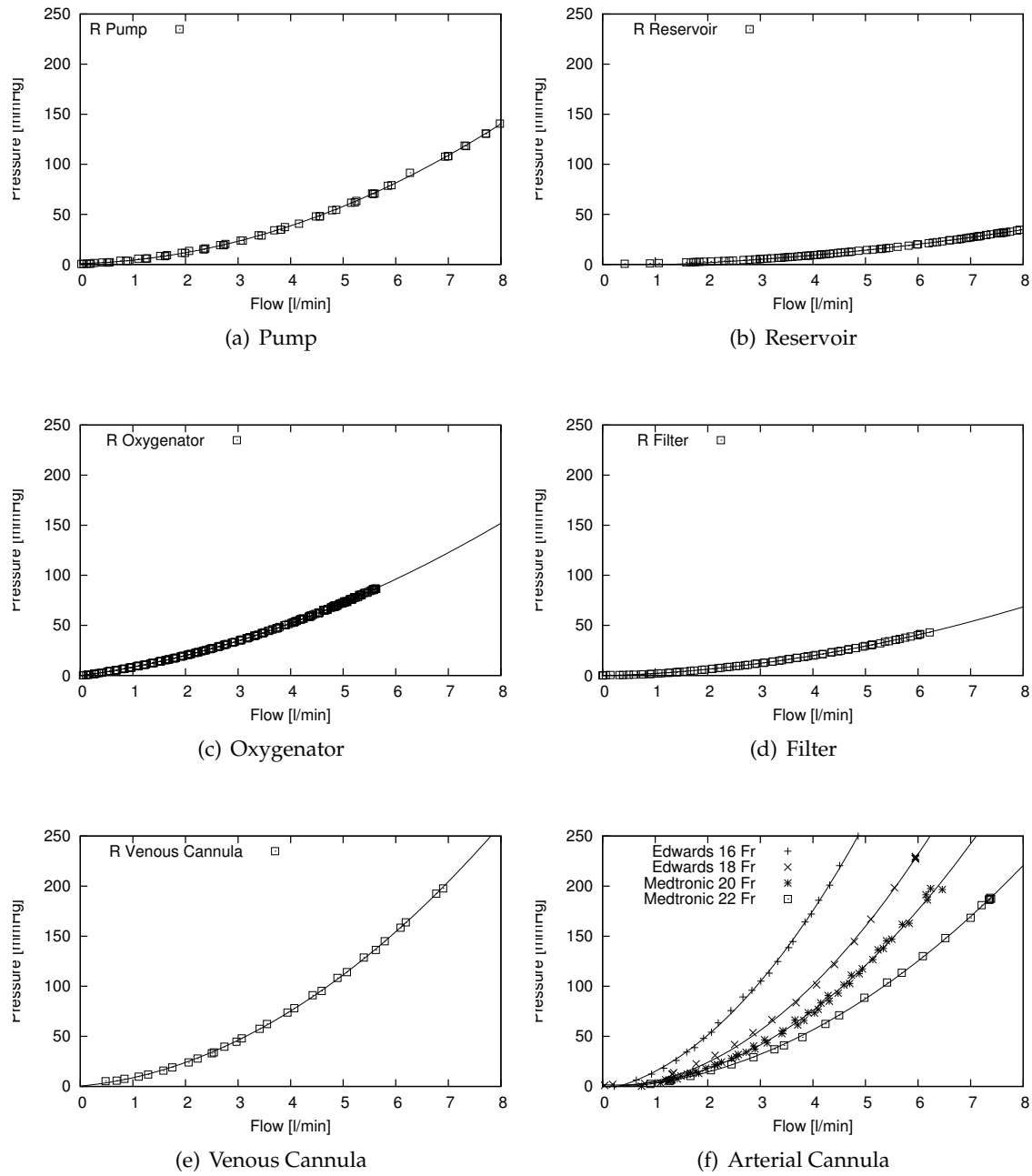


Figure III-22.: ECSS components

The constants K_a and K_b were extracted for each component obtaining the values as shown on table III.1.

i	Name	Ka_i	Kb_i
1	Pump Resistance	1.4	6
2	Reservoir	0.45	1.3
3	Oxygenator	0.9	10
4	Filter	0.7	2.7
5	Tubing	0	25
6	Venous Cannula	2.5	10
7	Arterial Cannula	3.8	6

Table III.1.: Component Constants

2.5.a. Centrifugal Pump

The centrifugal pump was modeled based on the assumption that the pump generates a constant head which depending on the output resistance will be converted to flow. This is expressed in equation (III.16)

$$H_{pump} = (p_2 - p_1)/(\rho g) + v^2/(2g) \quad (III.16)$$

where:

H_{pump} = total head developed [m]

p_2 = pressure at outlet [N/m^2]

p_1 = pressure at inlet [N/m^2]

ρ = density [kg/m^3]

g = acceleration of gravity (9.81) [m/s^2]

v = velocity at the outlet [m/s]

The velocity of the outlet is equal to the flow divided by the area at the outlet, as expressed in formula III.17.

$$\begin{aligned} v &= Q/A \\ A &= \Pi \cdot r^2 \end{aligned} \quad (III.17)$$

Several experiments were done to calculate the pump head at different pump speeds. The relation to flow was obtained by gradually changing the resistance at the outlet of the pump, starting from a completely closed state to a completely open state. The results are shown in figure III-23. To simplify the model calculations a linear approximation was considered to express the change between head and output flow for a given pump speed (RPM). This is expressed in formulas III.18 where $Pump_s$ expresses the static head for a given speed and $Pump_d$ (dynamic) represents the head decrease in relation to the flow.

$$\begin{aligned} H_{pump}(Q, w) &= Pump_d \cdot Q + Pump_s \\ Pump_d &= -7.8 \times 10^{-5} \cdot \omega + 2.2 \times 10^{-2} \\ Pump_s &= 2.5 \times 10^{-3} \cdot \omega - 2.7 \times 10^{-2} \end{aligned} \quad (III.18)$$

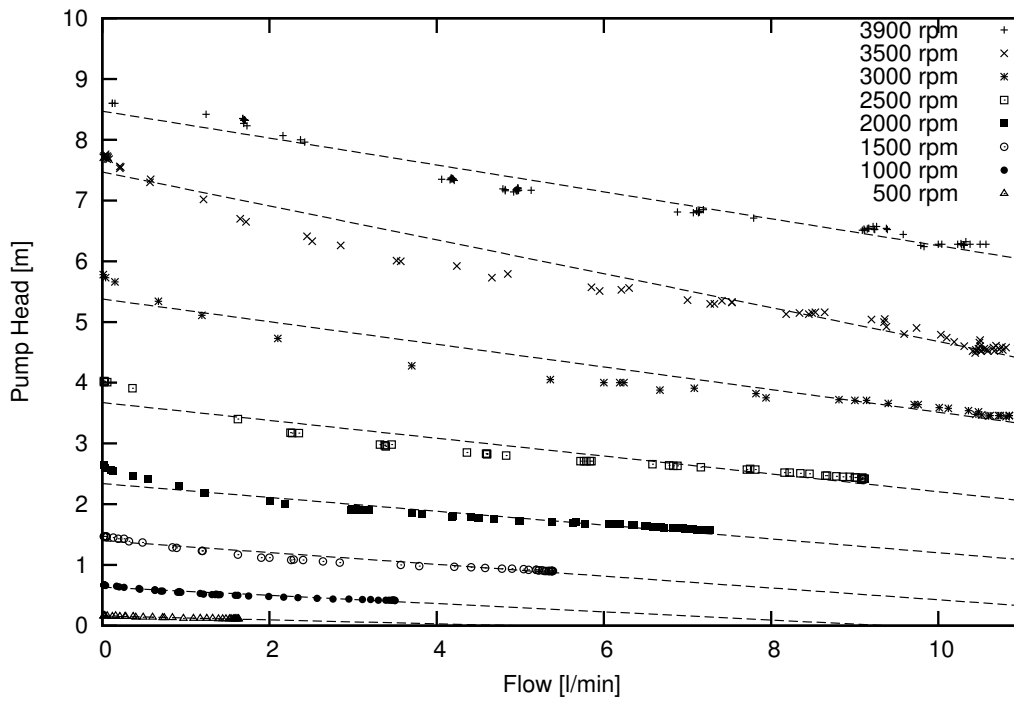


Figure III-23.: Pump Head Calculation

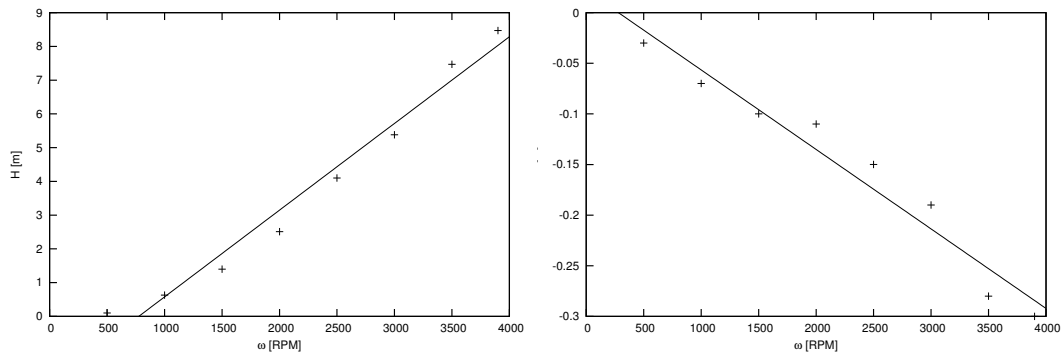


Figure III-24.: Pump static and dynamic constant approximation

The resulting flow generated by the ECSS may be calculated given the following formula:

$$\left(\frac{\sum_{i=1}^7 C a_i}{\rho g} + \frac{1}{2A^2 g} \right) Q^2 + \left(\frac{\sum_{i=1}^7 C b_i}{\rho g} - P_D(\text{rpm}) \right) Q - P_S(\text{rpm}) = 0 \quad (\text{III.19})$$

Where Q is the resulting flow, P_S and P_D are factors obtained depending on the rpm of the centrifugal pump and ΔP_{CVM} is the difference of pressure obtained from the input and output of the cardiovascular model. A is the pump output area, ρ is the blood density and g is gravity.

The rotational acceleration of the centrifugal pump was calculated by giving a maximum speed starting from 0 rpm. This was repeated with different resistances and also compared when the pump was at maximum speed and decreased to 0 rpm. From this results almost constant acceleration was detected of 214rpm/s and -239rpm/s for a decrease in speed.

Figure III-25 shows the result of the calculated model with all the components using two different arterial cannulas (System 1 with Edwards cannula and System 2 with Medtronic cannula). The system was evaluated by increasing the centrifugal pump speed starting from 0 up to maximum speed.

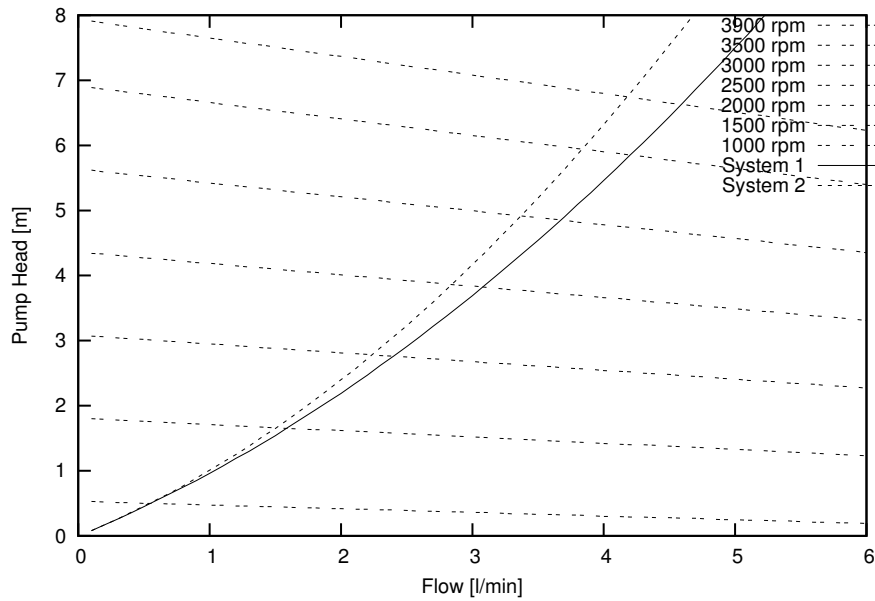


Figure III-25.: Complete Model

2.6. Blood Concentration and Medication Model

2.6.a. Vasoactive Drugs

The vasoactive drugs were modeled in terms half-times characteristics of each drug and their effects by changes in the vascular elements of the system (vascular resistance and

compliance). For our research Sodium nitroprusside (SNP) was modeled as a vasodilator and as vasoconstrictor Norepinephrine (NEP). Additionally volume may be added to increase the overall pressure. The medication model was designed based on the six compartment model explained in the gas exchange section (2.4). Drug distribution was calculated through the concentration change in each compartment.

The drug and volume elimination process was calculated from the concentrations within the capillary compliance considering the half life of each drug [61].

The effect of the vasoconstrictor and vasodilator was modeled with the following formula:

$$\begin{aligned} R_a &= R_0 + Ct_{a,d} \cdot Kr_d \\ C_a &= C_0 + Ct_{a,d} \cdot Kc_d \end{aligned} \quad (III.20)$$

Where R_a and C_x is the new compliance value of the arterial resistance and compliance respectively. R_0 and C_0 are the base values, $Ct_{a,d}$ is the drug concentration in the arterial compartment and Kr and Kc are constants for each drug.

2.6.b. Volume

The administration of volume was simulated by increasing the volume of the reservoir compartment in the ECSS. The same six compartment model was used to estimate the hematocrit (Hcrit) in the system. When the ECSS is not connected the compartments that correspond to the cardiovascular system start with a Hcrit of 45%. The volume contained in the compartments of the ECSS contains a Hcrit of 0%. When the clamps of the ECSS are opened the concentrations in the different compartments start to mix, reducing the overall Hcrit in the system. The volume abortion was modeled by reducing the volume in the capillary compartments considering a half life constant of the administered substance. This will make the Hcrit recover its initial value.

2.6.c. Viscosity

Previous studies have been carried out analyzing the effect of viscosity due to different hematocrit levels and changes in temperature[71]. The relation between viscosity, hematocrit level and temperature is shown in figure III-26.

From these results a function of viscosity in terms of temperature and hematocrit:

$$Viscosity = 0.0394 \cdot Temp - 1.22 + 0.127((19 + Hcrit)/Temp)^{3.88} \quad (III.21)$$

The value obtained from the previous formula was used to modify a resistance parameter in the systemic circulation.

2.7. Patient Disorders and Possible Complications

Several parameters were used in the model to represent the behavior of a patient with different disorders.

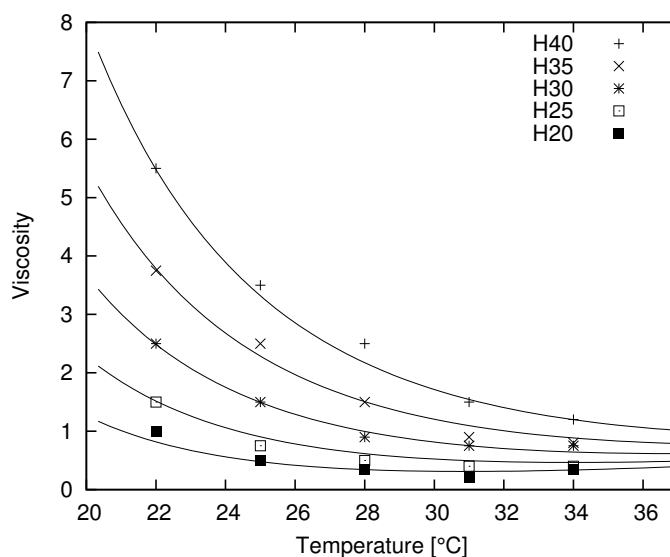


Figure III-26.: Viscosity vs Hcrit vs Temperature

Heart-Rate

The heart-rate may be increased producing tachycardia or decreased to produce bradycardia. Cardiac arrest may also be produced at any time producing heart full stop.

Heart-Valve Disorder

It is possible to simulate a malfunction of the mitral valve, causing a back flow between the left atrium and left ventricle of the heart.

Heart Elastance

Decrease of the elastance from the right or left ventricle may be simulated producing less cardiac output.

Hypertension and Hypotension

Hypertension and hypotension may be produced by changing the pulmonary resistance or the vascular resistances.

Blood loss and occlusion

In the ECSS it is possible to simulate leakage from a bad connection or the clamping of the arterial/venous tubing connecting the ECSS and the patient.

3. Simulation

This section describes how that model was used to construct a simulation environment that is capable of performing continuous simulation. From this system it is possible to extract parameters such as current MAP and EFR. These values will be later on sent to an external program in charge of automation, the control outputs are then used as inputs generating a continuous interaction with the model.

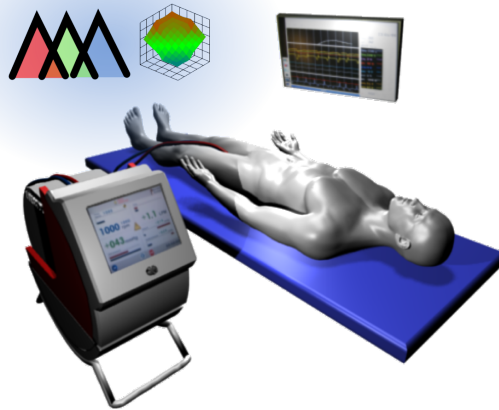


Figure III-27.: Patient Simulation under ECC

3.1. Simulation Environment

3.1.a. JSim

JSim was used as the main component of the simulation environment. JSim is a Java-based simulation system for building quantitative numeric models and analyzing them with respect to experimental reference data. JSim models may intermix ODEs, PDEs, implicit equations, integrals, summations, discrete events and procedural code as appropriate. JSim's model compiler can automatically insert conversion factors for compatible physical units as well as detect and reject unit unbalanced equations.[39, 74]

JSim model calculations are specified in JSim's own Mathematical Modeling Language (MML) an easy-to-read text-based language. MML models are most often expressed in terms of mathematical equations (for example, ordinary and partial differential equations, implicit equations). MML is constructed so that model writers may combine mathematics, events and procedural code as needed. JSim supports multiple "domains" or independent variables (e.g. time, space). MML variables may vary over any combination of the declared domains.

To execute a model a predefined time is set defining the length of the simulation. This time and simulation step is used to allocate the memory necessary to store the resulting values of all the variables in the model. Afterwards the contents of all the variables may be

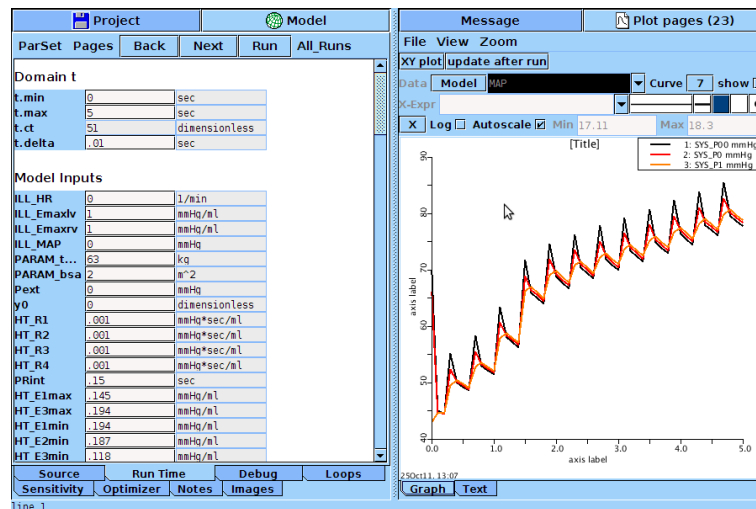


Figure III-28.: JSim

graphed and analyzed. JSim by itself is not capable of performing a continuous simulation, and for this reason a simulation manager was created.

3.1.b. Simulation Manager

The simulation manager takes control of JSim by first loading the required mml model saved in project files. From these files the different variables of the model are extracted and separated into the following groups:

- **State variables** These variables are the ones used by the mathematical solver. They define the current state of the model to start the simulation.
- **Control variables** These variables are all the parameters that may be set before the simulation is started, but remain constant during simulation.
- **Output variables** These are the variables that may be needed from the model to be used as control outputs or to be observed during simulation.

Simulation Frames A simulation frame was created to store the model information at a specific time within the simulation. It consists of a set of state variables, control variables and output variables. With this frames it is possible to reload a simulation model and start it at a previous state that requires constant test.

Simulation Engine The simulation engine enables the continuous simulation by using the described simulation frames. It first takes the state variable and control variables of the current frame and places them in the model. Then a predefined simulation time is executed with a step size. After execution is finished the last value of the state variables of the model are used to update the state variables of the active frame. Additionally the output variables of the frame are updated from the model. In the next step the new values

III. Experimental and Mathematical Models

of the state variables are introduced together with the control variables that have changed in relation to the previous simulation step. Once this is done the simulation is restarted.

3.1.c. User Interface

A graphical user interface was designed to take control of the simulation parameters. In a control panel the different parameters of the model may be modified. Additional overview and monitor panels show the different values of the model while the simulation is being executed (Figure III-29).

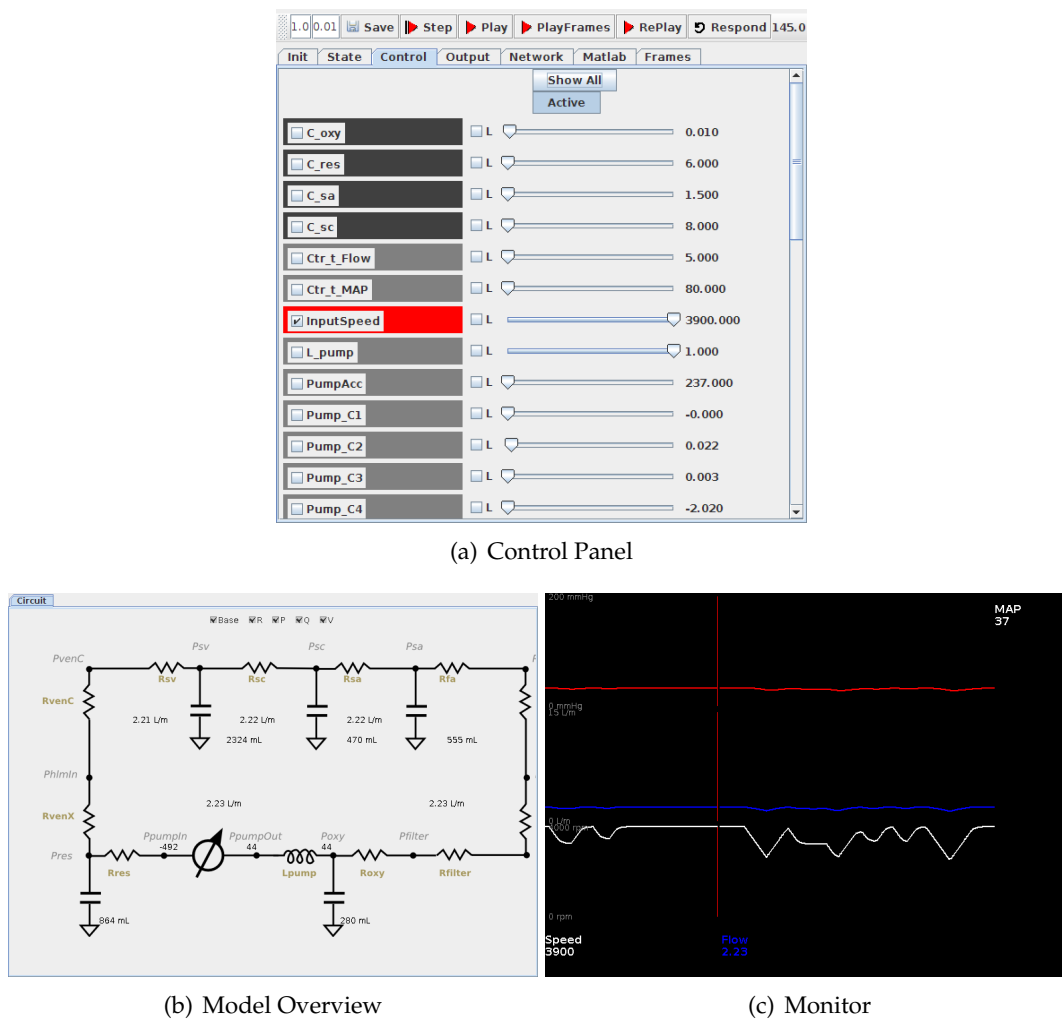


Figure III-29.: Simulation environment user interface

Figure III-30 shows the Simulation Frames panel where several parameters have been modified to simulate an operation procedure.

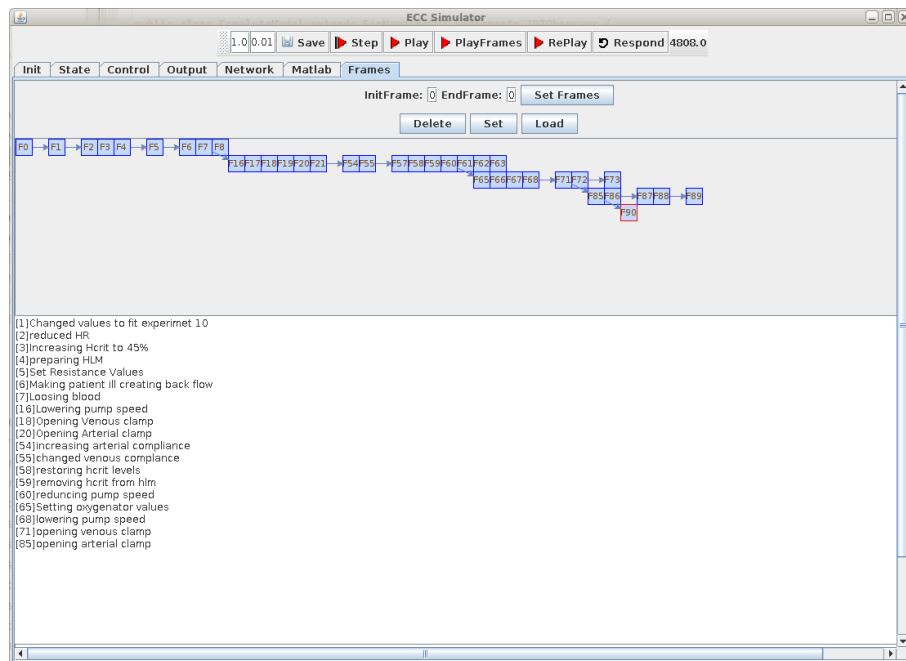


Figure III-30.: Simulation Frames

3.1.d. Simulation Interconnection

Figure III-31 shows the connection of the simulation environment to other components. This is done by using data packets that are sent between the different applications through a TCP/IP connection.

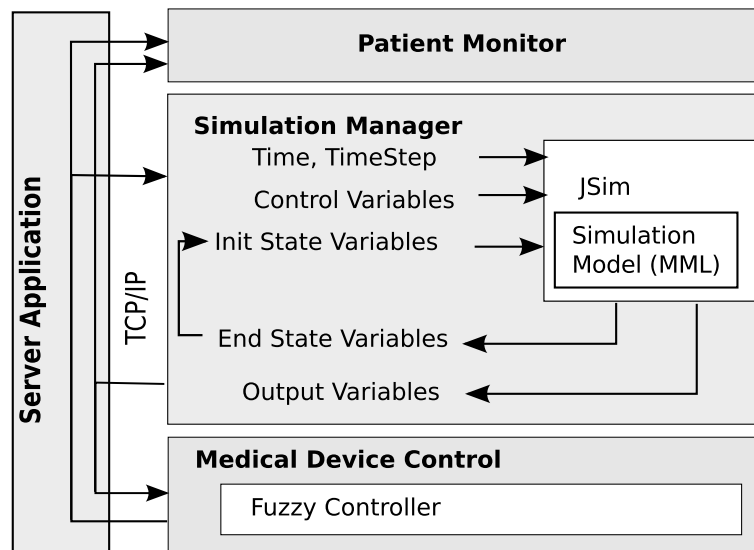


Figure III-31.: Simulation interconnection

The simulation manager takes out the output variable of the simulated model and con-

structs the output packet. This packet is sent to a monitor for visualization and the control module. The output of the controller are sent back and introduced in the model for the next simulation step. The next chapter will describe the fuzzy controller in section IV.1 and section IV.3 will describe the medical device control.

Additional information about the components of the simulation implementation may be found in the appendix in section B

3.2. Simulation Results

This section shows the simulation results after the model parameters were adjusted, compared with the experimental data obtained from the experiments described in section 1.

3.2.a. Arterial Pressure

Figure III-32 shows the comparison graph of the arterial pressure between the experiment measured at the aorta, and the mathematical model. To obtain this the heart rate of the model was set to the same in the experiment at 92 bpm. The systemic resistances were adjusted to generate the same systolic and diastolic pressure. The isovolumic relaxation produced by the closure of the aortic valve was simulated by a small back flow in the corresponding valve.

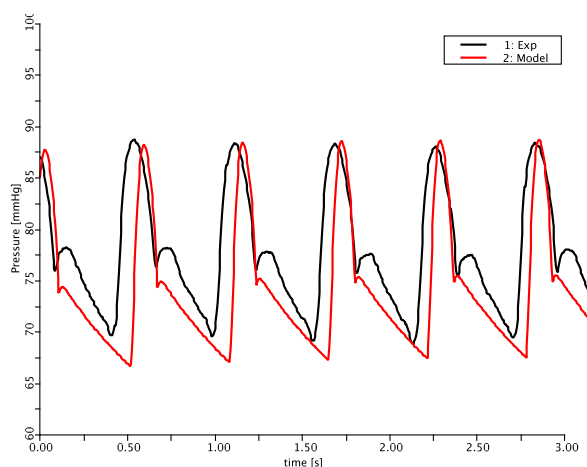


Figure III-32.: Arterial Pressure

3.2.b. ECSS Pressures and Flow

Figure III-33 shows a comparison between the experiment and simulation of the pressures and flows at different pump speeds. The values of the experiment are shown as dots whereas the simulation results are the continuous lines.

The pressures at the arterial and venous cannula together with the inlet and outlet of the ECSS had similar results to the experimental data. The mean arterial pressure was 10 mmHg lower at slow speeds and increased almost linearly when reaching higher speeds. The results from this particular experiment were different showing a smaller increase of

pressure with higher speeds. The EFR however corresponded to that obtained experimentally and both the flow at the aorta ascendens and descendens where decreased accordingly, showing slight variations at around 2000 rpm.

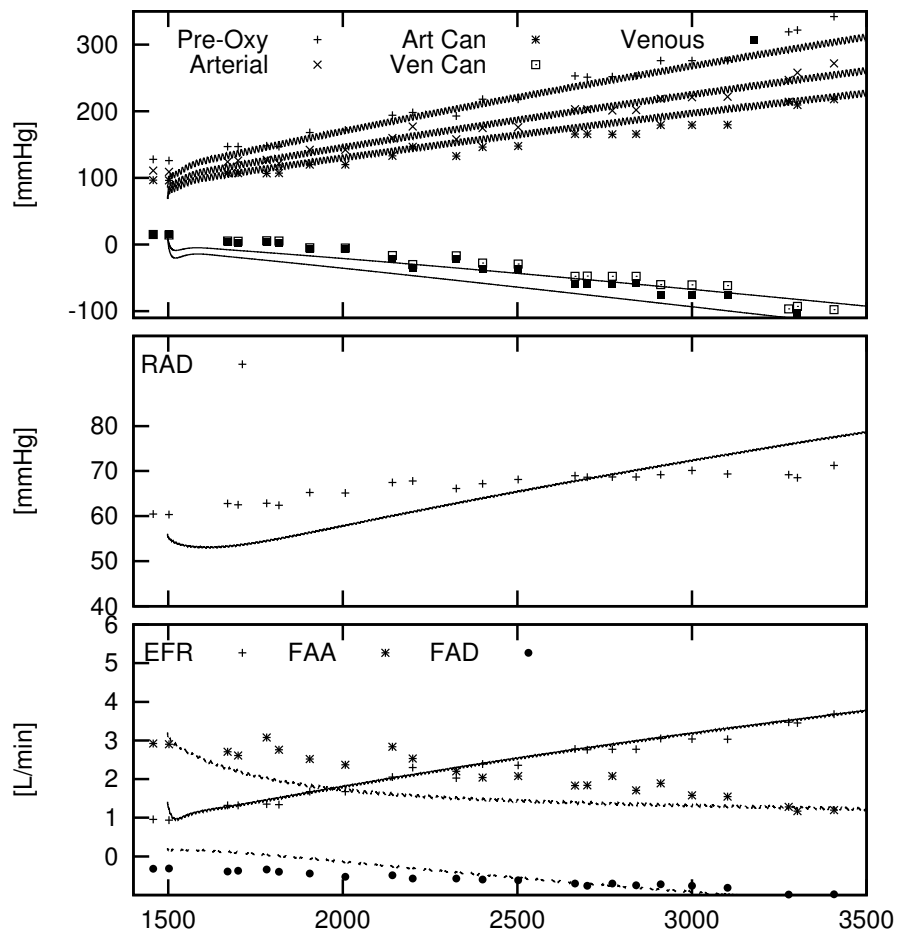


Figure III-33.: ECSS at different RPM.

3.2.c. Medication

Figure III-34 shows the simulation results when a vasoconstrictor substance was introduced into the system. The bottom graph shows the concentration change in the different compartments. These compartments correspond to the reduced 6 compartment model used for the vasoactive substances (figure III-20).

The vasoconstrictor substance was introduced into the reservoir compartment of the ECSS and from there it was spread to the rest of the compartments. The effect of the heart rate and the increase of resistance/compliance was calculated from the amount of medication in the arterial compartment. Due to the high resistance in the capillaries the medication that arrives to the venous compartment is at a much slower rate. After a certain amount of time the concentration of the medication becomes uniform and slowly decreases according to the half-life of the vasoconstrictor.

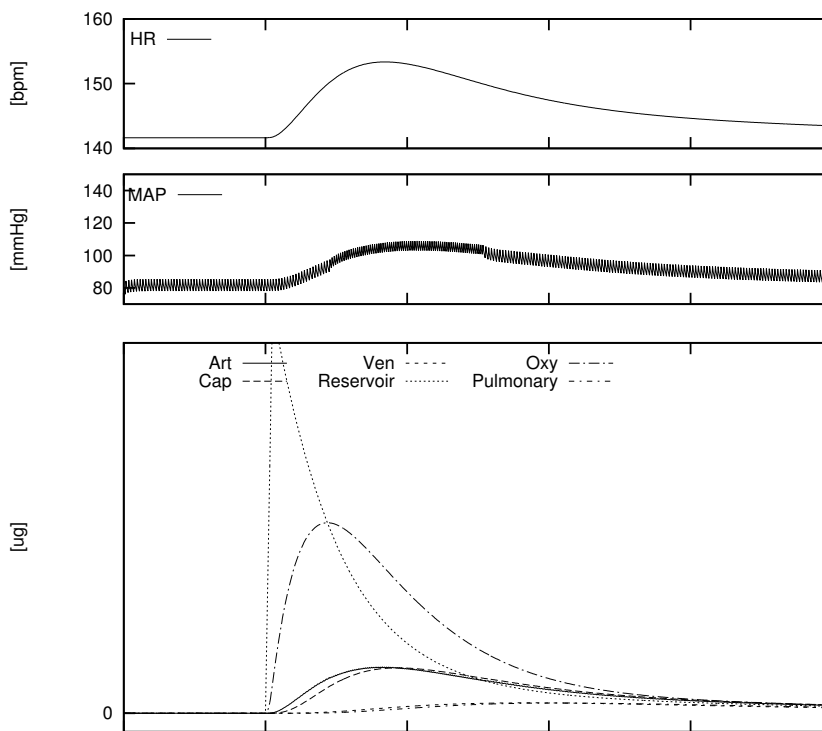


Figure III-34.: Medication concentration distribution in different compartments.

Vasoconstrictor Figure III-35 shows the comparison between the simulation and experiment when a dosage of NEP as a vasoconstrictor was administered. The model was set to start at the same MAP as the experiment. The results show that both had a similar peak mean arterial pressure. From the experimental results a significant decrease of pressure may be observed after 2 minutes, looking at the additional recordings it was noticed that

this was caused by an internal mechanism of the ECSS that opened an internal clamp generate a momentaneous circulation of blood in the system to avoid clotting, thus affecting the pressure of the hole system. After the clamp is closed again the pressure is recovered and the effect of the drug continues its normal effect. The model does not account for this change of pressure and continues smoothly.

The effect of the drug is reduced in a similar manner to both graphs showing that in the model the arterial pressure was reduced more than in the mathematical model.

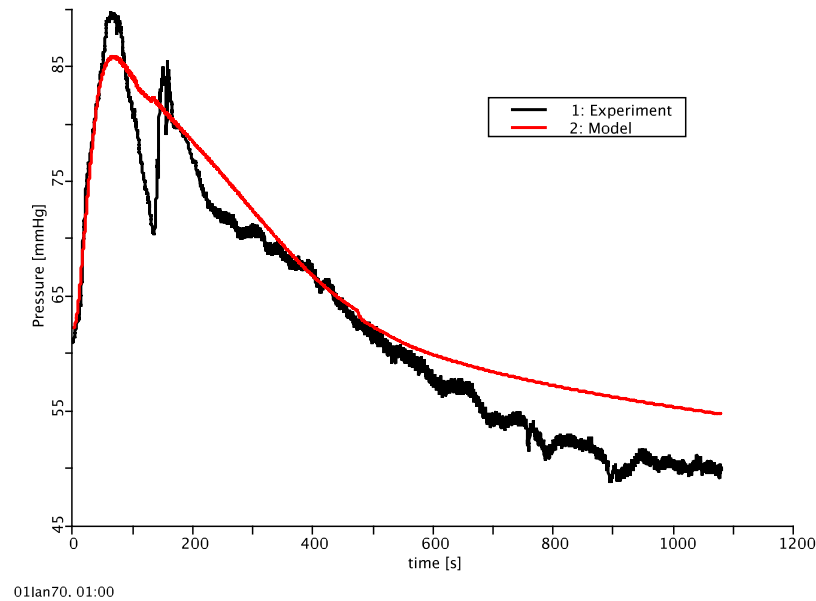


Figure III-35.: Vasoconstrictor effect in simulation model and experiment.

Vasodilator Figure III-36 shows the result for the contrary effect of the previous drug; a vasodilator using ISDN. The mean arterial pressure was reduced for both cases. In the experiment data when the effect of the drug was reduced and the mean arterial pressure was rising we saw a similar effect as the vasoconstrictor, a sudden drop of pressure. This as mentioned before corresponds to the internal clamps of the ECSS opening and closing to avoid clotting within the system components.

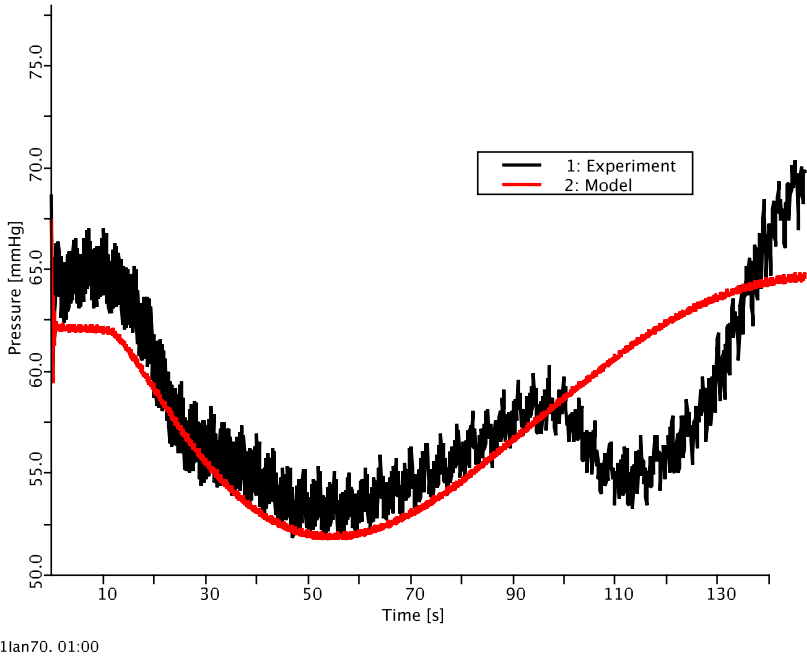


Figure III-36.: Vasodilator effect in simulation model and experiment.

3.3. Conclusion

The experimental model described in this chapter was valuable for the acquisition of data that could be later used as reference for the construction of the mathematical model.

These results show that the mathematical model is capable of representing the experiment model in terms of inlet and outlet pressures, MAP, EFR and effects of medication.

The interaction between the flow generated by the heart and the change in pressure was noticed to be different from one experiment to another and may require additional adjustments in the model. However this also shows a difficulty in designing a controller and optimizing it for a specific case, bringing up the need of an adaptive mechanism that will be explained in the following part of this thesis.

Resistance Analysis

Further study of the constructed ECSS model allowed the analysis of how much resistance was produced by each individual component. From these results and the characteristics of the pump it is observed that the pump is capable of producing enough flow, however with a high resistance at the outlet this flow is reduced. If the EFR needs to be higher than what was obtained experimentally then several components may be adjusted changing the components with a higher resistance. A suggestion of which components to consider may be obtained from figure III-37.

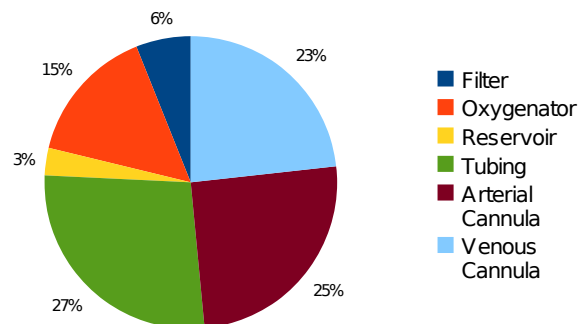


Figure III-37.: Component resistance comparison.

Automation of the Extracorporeal Circulatory Support System

Contents

1. Fuzzy Control	75
2. Multiple Input Adaptive Fuzzy Controller	82
3. System Implementation	88
4. Portable ECSS Control	102

This chapter focuses on the aspects that will enable the automation of the portable ECSS. Section 1 introduces the fundamentals of fuzzy logic giving a brief history and describing its basic components. This is then extended to the description of an adaptive mechanism which is based on reference models. The described controller is then implemented in section 4 for specifically the ECSS. The last section of this chapter will then show the hardware implementation of the designed automation system.

1. Fuzzy Control

Fuzzy logic was introduced by Zadeh in the 1960s[99] and since then it has extended its use in various disciplines such as control theory and artificial intelligence. Fuzzy logic has a particular advantage in areas where precise mathematical description of the control process is not possible. It provides a formal methodology for representing, manipulating and implementing a human’s heuristic knowledge about how to control a system.

Fuzzy logic is based on representing the inputs and outputs in a linguistic manner and describe in the form of rules how the current value of these inputs should affect the outputs. As an example in the speed control of a pump to get a desired flow, the amount of flow may be represented as “Low”, “Normal” and “High” compared to a target value. The pump speed may be changed in the forms of “Increase”, “Steady” for no change and

“Decrease”. Then rules may be created to control the pump speed in the following way:

IF Flow = “Low” THEN Speed = “Increase”
IF Flow = “Normal” THEN Speed = “Steady”
IF Flow = “High” THEN Speed = “Decrease”

In fuzzy logic these linguistic terms that represent each input and output are described as fuzzy sets.

1.1. Fuzzy Sets

Fuzzy sets are represented by membership functions. They are defined in a universe of discourse, which is the range of values in which the fuzzy variables are expected to be. The membership function gives the grade, or degree, of membership within the set, of any element of the universe of discourse. The membership function maps the elements of the universe onto numerical values in the interval [0, 1]. A membership function value of zero implies that the corresponding element is definitely not an element of the fuzzy set, while a value of unity means that the element fully belongs to the set. A grade of membership in between corresponds to the fuzzy membership to set. These fuzzy sets are used to represent the inputs that will define the controller and the outputs. As an example figure IV-1 shows the fuzzy description of a variable by using three membership functions representing the range of “Low”, “Normal”, “High”. When the value of this variable is of 1 it is considered as “Low” with a degree of membership of 1. If the value is 1.5 the degree of membership will be 0.5 of “Low” and 0.5 of “Normal”.

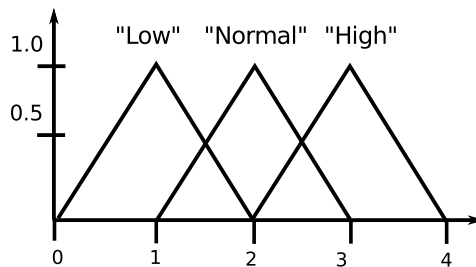


Figure IV-1.: Membership Function

The linguistic terms describing each input and output may be increased to describe in a more detailed manner how the controller should act. For example flow could be described as “Very low”, or “Very very low” and the output could be described as “Slight Increase” or “Big Increase”. To simplify this matter and make it more clear to visualize these linguistic terms where modified by a numerical representation as in table IV.1. Their representation may then be easily extended depending on the number of fuzzy sets that are to be used for each variable, however it should be kept in mind that these numbers will not represent the actual value of the variable but the set to which it is referring, similar to a grading system.

Input	Output	Equivalent
"Very Low"	"Big Decrease"	-2
"Slightly Low"	"Small Decrease"	-1
"Normal"	"Steady"	0
"Slightly High"	"Small Increase"	+1
"Very High"	"Big Increase"	+2

Table IV.1.: Representation of inputs and outputs

1.1.a. Fuzzy Set Distribution

The fuzzy sets may be distributed in different ways to cover all the universe of discourse. First by defining the number of sets, where the middle set may represent "0" for variables covering positive and negative values, or by having sets only in the positive side. A second feature is the amount of overlap between sets. If the overlap is zero then for each variable there will be at most one active set. If the overlap is to 50% then one set will cover half of the neighbor sets. This may be seen in figure IV-2. Another feature in the distribution of sets is if the sets are equally distributed along the universe of discourse, or when the sets become closer together when they are close to zero. This is depicted in figure IV-3.

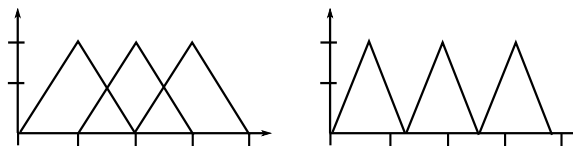


Figure IV-2.: Overlap configuration in fuzzy sets

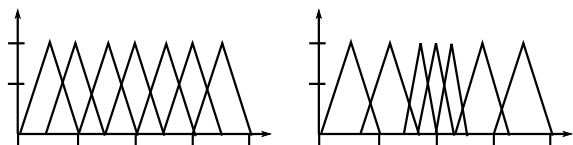


Figure IV-3.: Focus vs regular distribution

1.2. Fuzzy Control Structure

A fuzzy controller is composed of the following:

1. The *fuzzification* makes a conversion between the analog input variables into a linguistic representation, with the use of fuzzy sets.
2. A *rule base* holds the knowledge in the form of a set of rules, of how best to control the system.

3. An *inference mechanism* which evaluates what control rules are relevant at the current time and decides what the input to the plant should be.
4. A *defuzzification interface* converts the conclusions reached by the inference mechanism into the inputs to the plant. Two major methods are used for defuzzification: The maximum defuzzification method, where the output value is determined by the linguistic variable with the maximum weight; The centroid calculation defuzzification method, where an output value is determined by the weighted influence of all the active output membership functions.

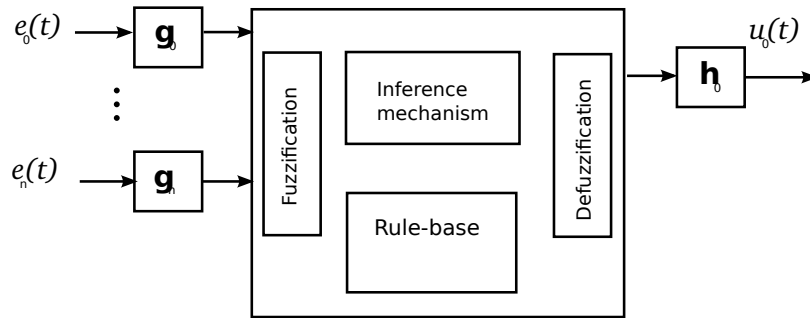


Figure IV-4.: Basic Fuzzy Controller

This basic structure was extended introducing input and output gains. These gains are used as an interface between the external world and the controller. This means that the controller will work internally in the range of -1 to $+1$. In case of the inputs the real value is multiplied by the gain g_i . In the case of the normalized output the value is divided by the gain h_o .

The rule base consists of k linguistic control rules R_1, \dots, R_k that are of the following form:

$$R_r : \mathbf{IF} \ x_1 \ \text{is} \ A_1^l \ \mathbf{and} \ \dots \ \mathbf{and} \ x_n \ \text{is} \ A_n^l \ \mathbf{then} \ y \ \text{is} \ G^l \quad (\text{IV.1})$$

Where $r = 1, \dots, k$ for the number of rules in the rule base, A and G are linguistic terms of input and output variables, respectively. x and y are the input and output of the fuzzy logic system.

Controllers operating on such rules are called *fuzzy controllers*. There are several types of controllers defined by the interpretation of a certain rule and of the rule base. The most common types are the *Mamdani controller* and the *Sugeno controller*. This thesis focuses on the first type of controller which is described in the following subsection.

1.3. Mamdani Controller

A Mamdani controller combines a given input rule with the rule antecedents by determining the degree to which each input belongs to the corresponding fuzzy set. For a rule to be active all linguistic terms in the antecedent of a rule must be fulfilled, these values are conjunctively aggregated in a suitable way:

$$\tau_r = \min\{\mu_{x,r}^{(1)}(x_1), \dots, \mu_{y,r}^{(n)}(x_n)\} \quad (IV.2)$$

Where τ_r is the *degree of truth* of the antecedent of rule R_r with the given inputs (x_1, \dots, x_n) . μ represents the active sets of each corresponding input. The combination of degree of truth of all the rules in the rule base is calculated by a max operator:

$$\epsilon_{x_1, \dots, x_n}^{output} = \max_{r \in \{1, \dots, k\}} \min\{\mu_{x,r}^{(1)}(x_1), \dots, \mu_{y,r}^{(n)}(x_n), \epsilon(y)\} \quad (IV.3)$$

Figure IV-5 shows a representation of these operations for two rules, obtaining the output set ϵ .

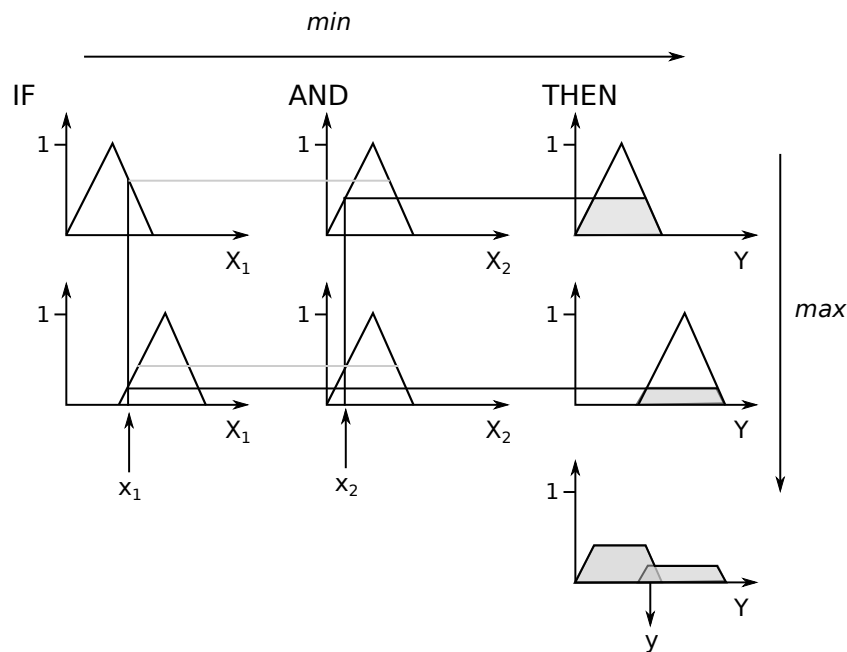


Figure IV-5.: Rule evaluation with min max operators

The last operation to obtain a crisp output is to convert the ϵ output into a crisp value corresponding to y output. There are several defuzzification methods two of them are described as follows:

1.3.a. Mean-of-Maximum Method (MOM)

This method is the average of all the values where the fuzzy set ϵ assumes a maximum membership degree. MOM can result in a discontinuous change of the output variable.

1.3.b. Center-of-Gravity Method (COG)

This method calculates the average of all possible output values weighted by the degree of membership of the fuzzy set ϵ and adds these values. Then the result is divided by the sum of all membership degrees.

The center-of-gravity gives a smother response than the MOM procedure and is the most widely used. The fuzzy controllers used in this work use the COG as a defuzzificaton method.

1.4. PI and PD Fuzzy Controller

By using a fuzzy controller it is possible to construct equivalent conventional Proportional Derivative (PD) and Proportional Integral (PI) linear type of controllers. This type of control has been previously described by Li [79]. This allows a more systematic creation of rule bases. If we consider the first example of controlling a pump to reach a specific flow and we also introduce the change of flow over time then we would have rules such as:

IF Flow = "Low" AND Flow "Decreasing" THEN Speed = "Big Increase "
 IF Flow = "Low" AND Flow "Stable" THEN Speed = "Increase "
 IF Flow = "Low" AND Flow "Increasing" THEN Speed = "Small Increase"

The result of creating all the combinations for the previous case is the equivalent of using a FuzzyPI controller. However by understanding the structure of this type of controller the amount of input and output sets may be easily expanded to create more precise rule sets.

The FuzzyPI and FuzzyPD controller has as input signals the error (e) and the derivative of the error (\dot{e}).Using the same notation as what was described in table IV.1 the rule base of a Fuzzy-PI controller is built as follows:

The number of sets for the output variable to cover the complete range of combinations is calculated with the following formula:

$$NumberOutputSets = numInputs * granularity - (numInputs - 1) \tag{IV.4}$$

For this case all the inputs are considered to have the same granularity. The rule base for the FZ-PI will consist of a diagonal of set '0' , passing through the center of the rule base where there is total equilibrium and the controller has reached its target value ($e = 0, \dot{e} = 0$) or when e is compensated by \dot{e} to reach the target value.

The rule structure of the Fuzzy-PI and Fuzzy-PD are shown in the following tables:

\dot{e} / e	-3	-2	-1	0	+1	+2	+3
-3	+6	+5	+4	+3	+2	+1	0
-2	+5	+4	+3	+2	+1	0	-1
-1	+4	+3	+2	+1	0	-1	-2
0	+3	+2	+1	0	-1	-2	-3
+1	+2	+1	0	-1	-2	-3	-4
+2	+1	0	-1	-2	-3	-4	-5
+3	0	-1	-2	-3	-4	-5	-6

Table IV.2.: Rule base of a Fuzzy-PI

\dot{e} / e	-3	-2	-1	0	+1	+2	+3
-3	+3	+3	+3	+1	-1	-1	-1
-2	+3	+3	+3	+1	-1	-1	-1
-1	+3	+3	+2	+1	-1	-1	-2
0	+3	+2	+1	0	-1	-2	-3
+1	+2	+1	+1	-1	-2	-3	-3
+2	+1	+1	+1	-1	-3	-3	-3
+3	+1	+1	+1	-1	-3	-3	-3

Table IV.3.: Rule base of a Fuzzy-PD

Figure IV-6 shows a 3D representation of the surface generated by each type of control, having at the x axis the error, the y axis the derivative of the error and on the z axis the

output of the controller. The presented surface was created using sets that overlap each other. If there is no overlap in the sets the output surface will not be as smooth and will present steps changing from one set to another.

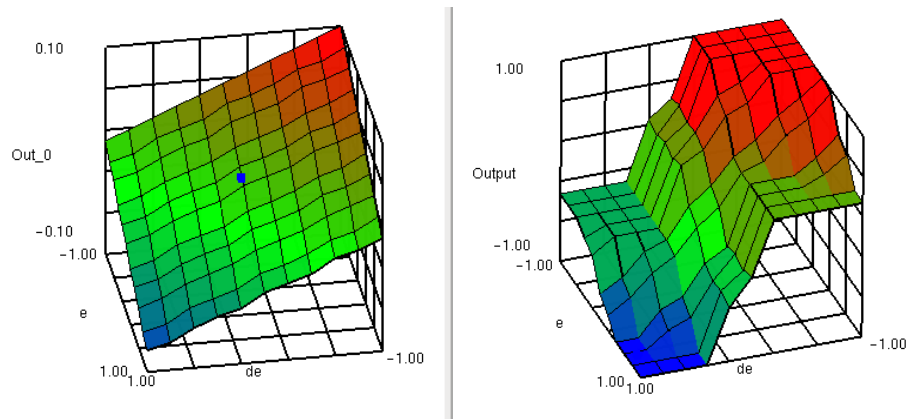


Figure IV-6.: Fuzzy PI and PD Control 3D Surface (normalized inputs [-1 1])

The following section (IV.2) will use as a base these type of controllers and afterwards implement an adaptive mechanism to modify the rule sets.

Error Definition

In classic control the error is defined by the set point (target) minus the output of the plant.

$$Error = Target - Output \quad (IV.5)$$

As an example if the set point is 80 mmHg and the plant output is 60 mmHg then the error will be 20mmHg. For a proportional controller this would mean that it needs to increase the output with a certain K_p which will increase for example the pump speed so the plant reaches the set point of 80 mmHg. For the fuzzy controller used, the error is calculated as the following:

$$Error = Output - Target \quad (IV.6)$$

For the previous example the resulting error will give -20 mmHg which would be understood as the plant having a lower pressure than the desired target by 20 mmHg. This can then be processed with the presented rules to make the appropriate corrections.

2. Multiple Input Adaptive Fuzzy Controller

The previous section described the principles of fuzzy logic. A fuzzy controller may consist of more than one input, however with the introduction of more inputs the creation of rules may become complex if every combination of sets needs to be considered. This section proposes a method to deal with multiple inputs, allowing the use of prior knowledge of experienced staff, while still maintaining a simple system. As a medical device that is used for the treatment of patients the conditions to control are highly dynamic and may change from one patient to another. The created rules are based on assumptions and previous knowledge of the expert. Further control adaptation is needed to adjust to the specific patient characteristics and to improve control performance. This section describes such adaptive mechanism.

Figure IV-7 shows the proposed control architecture. This consists on the inputs of the controller and target values defined by the operator; the control mechanism is based on a type of model reference control further explained in more detail. A reference error ($rError$) and a target error ($tError$) is calculated in a first stage. These two errors are used as inputs for two types of controllers; the first defined as a set of *knowledge controller* and the second as a single *adaptive controller*.

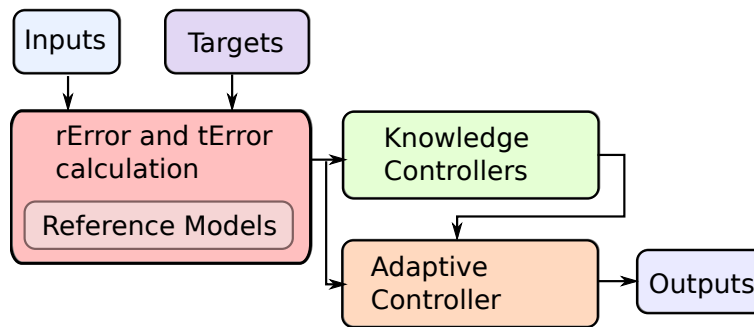


Figure IV-7.: Adaptive control architecture.

During the automation process, when the current inputs are introduced into the adaptive controller and there exist no rule exceeding a predefined threshold a *learning mechanism* takes place where the learning controllers are activated and a new rule is created. If a rule exists an *adaptive mechanism* is executed to modify the existing rules and improve control performance. The output obtained from the adaptive controller is then used to increase or decrease the value of the control variables. The following subsections will explain these processes in more detail.

2.1. Model Reference Control

The concept of model reference control together with fuzzy control has been previously proposed by several authors[100, 101]. The idea behind this type of control is to compare the output of the system to a desired response from a reference model. The control parameters are updated based on this error through an adaptive mechanism. The goal is for the parameters to converge to ideal values that cause the plant response to match the response of the reference model.

A second order function was used as a reference signal where the stabilizing time may be specified. When the controller is activated the reference signals are set to the current value of each parameter. With the reference model the following signals are generated: A target error (Te) which refers to the difference between the input signal and the target and a reference error (Re) which refers to the difference between the input signal and the reference model being calculated. This is depicted in figure IV-8.

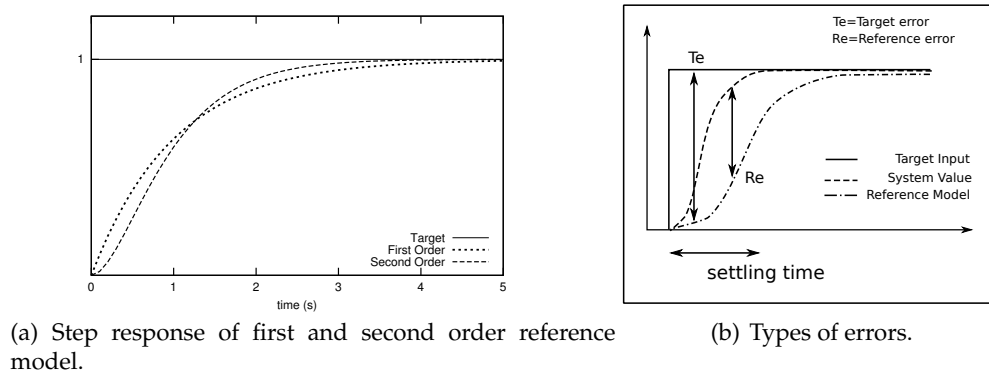


Figure IV-8.: Reference signal.

2.2. Knowledge Controllers

To deal with multiple inputs, while not increasing the complexity for the creation of rules, smaller controllers are defined containing rules of how the controller should react to only specific inputs. This forms the knowledge base of the control for a single output. The result of all of these controllers is then added to form a single output signal. Predefined rule structures can be used to easily construct these smaller controllers, for example a Fuzzy-PI type of controller may be used for one input, using its error and change of error.

2.3. Adaptive Controller

The previously defined knowledge controllers may not be optimally tuned for all the possible scenarios, therefore an adaptive mechanism is needed capable of recognizing when a change must be made and how the rules may be modified. It is difficult to detect what rules should be modified in each small controller, this was solved by introducing an empty fuzzy controller that can use the information from the knowledge controllers and create its own rules. This controller is referred to as the adaptive controller. The adaptive controller uses the same inputs as the knowledge base controllers together with additional inputs that may help to characterize a specific situation.

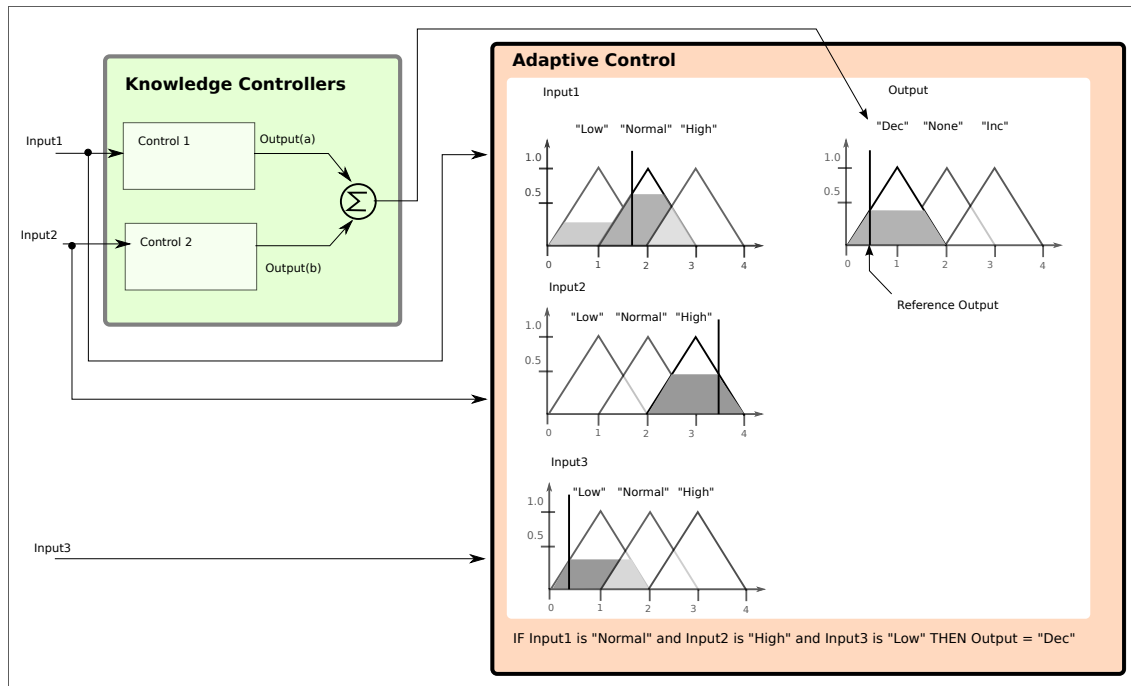


Figure IV-9.: Learning mechanism from knowledge controllers.

2.4. Learning Mechanism

The online learning mechanism is described in figure IV-9 and consists on the following steps:

1. tE and rE are obtained and the reference model is activated, on control start the reference model is set to start on the same value as the controller inputs. Afterwords the time step is continuously increased in the reference model.
2. The reference errors and derivative are introduced into the adaptive controller and knowledge controller.
3. The outputs of the knowledge controllers are calculated and the outputs corresponding to the same output parameter are summed to form a single output.
4. In the adaptive controller the sets with the highest degree of truth are extracted for each input; the degree of truth of a set is defined between 0 and 1, where the truth of 0 corresponds to the value of the input that is outside of the boundaries of the individual set, a degree of truth of 1 is given when the value of the input is the same as the maximum point of the set, the degree of truth in between is defined depending on the shape of the set (triangular, trapezoid, gaussean).
5. The new rule is created in the adaptive controller with the sets found in the previous step as antecedents and the summed output from the knowledge controller is used to find the set from the output variable with the highest degree of truth. The new rules are assigned with a weight of 0. This weight is later used in the adaptive mechanism.

We can observe from the learning figure IV-9 that when the new rule is created there might be a difference from the output obtained from the base controllers than from the output obtained from the new rule. This may be reduced by increasing the number of sets used for each variable in the adaptive controller.

2.5. Adapting Mechanism

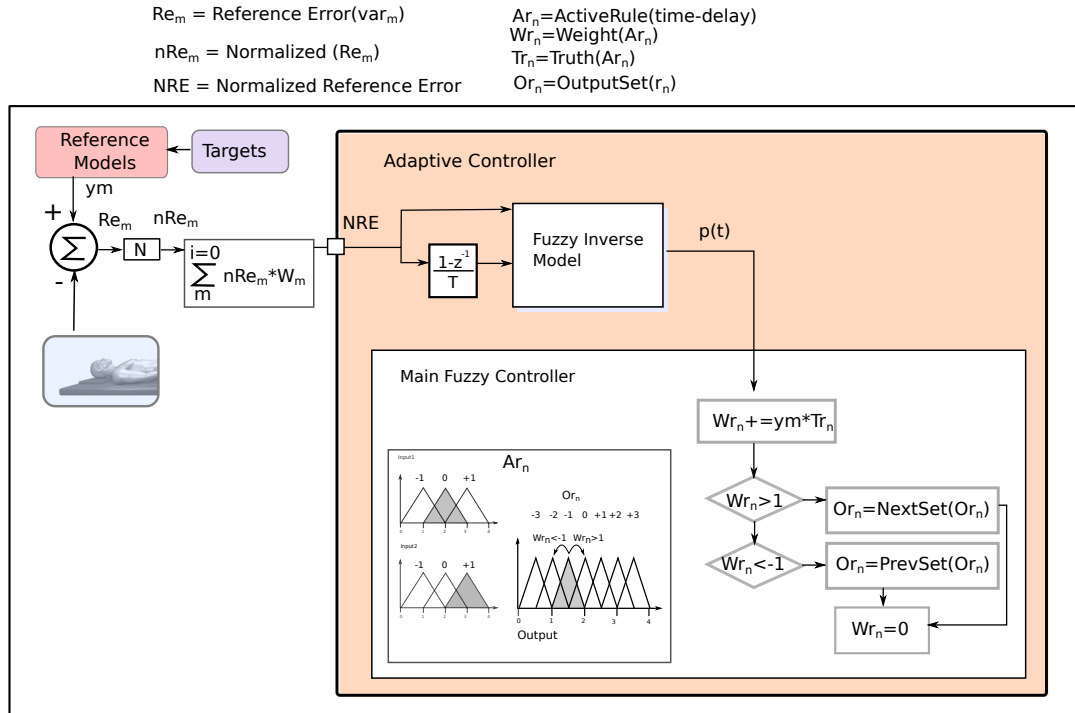


Figure IV-10.: Adaptive Mechanism.

The adaptive mechanism modifies the rules created from the learning process with the following steps:

1. The adaptive mechanism saves the active rule in each step. For each rule a weight is assigned, starting with 0 when first created.
2. A total normalized reference error (NRE) is calculated from all the input signals, consisting in the following:

$$\eta Re_i = \frac{Re_i}{T_i - v_{min,i}} \tag{IV.7}$$

$$NRE = \frac{\sum_{i=1}^n \eta Re_i \cdot w_i}{\sum_{i=1}^n w_i} \tag{IV.8}$$

Where ηRe_i is the normalize reference error, Re_i corresponds to the reference error, T_i is the target and $v_{min,i}$ is the minimum expected value. , all of $var i$. NRE adds all of the normalized reference errors multiplied by its individual weight and divided

by sum of all the weights. This normalization allows the possibility of using input signals of different nature such as pressure and flow and still be able to compare in an equitable manner how far are the input signals from their corresponding reference model.

3. After a predefined delay has passed the *NRE* and its change over time is used as input for a *Fuzzy Inverse Model (FIM)*. This model generates the signal $p(t)$ indicating how the weights of the active rules should be modified.
4. For the rules active at *time – delay* the rule weight is added with the output of the inverse control ($p(t)$) multiplied by its rule truth.
5. Rules are modified in the adaptive controller; if the weight of a rule exceeds +1 then the output set of the current rule is modified to use the next higher continuous set. If the rule weight is lower or equal to -1 the output set is set to the next lower continuous set. When the rule is changed the weight is set back to 0.

Rule delay definition

The delay used in step 4 will be important for the adaptive mechanism since it will define which rules to modify. If the system produces oscillations this delay is modified depending on the period of the oscillations; the delay should be established at most one fourth of the period of such oscillations.

Fuzzy Inverse Model

As described in step 3 this model determines how the rule weights should be changed and subsequently how the output sets for a specific rule are also changed. The *FIM* consists of a fuzzy controller with a rule structure equal to a Fuzzy-PI as described in table IV.3. This means that if for example the controller generated too much correction then the output of the *FIM* will be to reduce the weight for the rules at *time – delay* to produce less correction the next time these rules are used.

2.6. Performance evaluation

A mean absolute error (MAE) is used to calculate the performance of the controller, this is done using the sum of the normalized reference errors as shown in equation IV.9.

$$MAE = \sum_{i=0}^n \eta Re_i \cdot w_i \quad (IV.9)$$

Section V.1 of chapter V will give a comparative analysis of the difference of using a regular fuzzy controller, with the learning and adaptive mechanism.

2.7. Fuzzy Supervisor and Diagnosis

The properties of fuzzy logic also provide a tool to create rules specifying the conditions that represent the system working in a normal state or the conditions of failure. It may

be constantly monitoring all the signals in the system and identify what type of failure is occurring. As an example part of the experiment protocol in chapter III section 1.4.g specified the conditions in which the arterial and venous cannula were occluded. The venous cannula occlusion may be specified with the following rules:

IF PIn = "0" AND POxy = "+3" AND EFR="0" THEN ERROR="Arterial Occlusion"
IF PIn = "-3" AND POxy = "+1" AND EFR="0" THEN ERROR="Venous Occlusion"

Another use for fuzzy controllers is to describe what is happening in the system in a more verbal way and give a more clear understanding, rather than just presenting numbers. This is possible by modifying the fuzzy interface, and instead of giving a value at the output it may give the name of the active set and the truth value corresponding to the outputs. As an example the current value of MAP may be described as a range representing hypertension, another as normal and as hypotension. Triangular sets are used, with the top value at the lowest, middle and highest point respectively. Then simple rules such as the following:

IF MAP = "Low" THEN Pressure = "Hypotension"
IF MAP = "Medium" THEN Pressure = "Normal"
IF MAP = "HIGH" THEN Pressure = "Hypertension"

The output of this controller may not only specify the pressure type but also how much truth is in this value. More complex conditions may be created by using more inputs similar to the supervisor.

3. System Implementation

This section describes the design of an application framework created with the purpose of providing the necessary components for the automation of the ECSS. The objective of this framework however is not only focused on the ECSS but also to bring the possibility of using the same framework for the automation of other medical devices. In addition of a set of libraries, the framework consists of a set of tools that may facilitate the process of design, creation and verification of an automation system, decreasing the time of development and giving the developer the chance to focus more on the over-all requirements of the system rather than low-level system implementation. Some low-level software may still be required for non pre-existing systems, however the integration to the rest of the system is easy and once implemented this may be easily used by other applications.

3.1. System Description

Figure IV-11 shows an overview of the different components that are needed for the automation of a medical system.

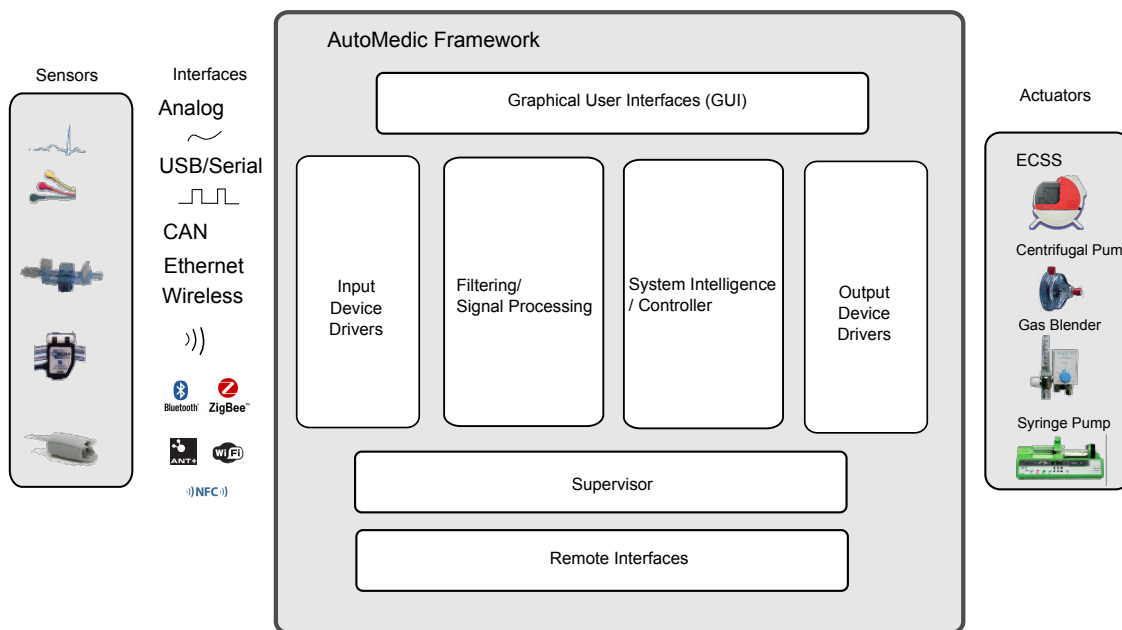


Figure IV-11.: System Components

Sensors In order to give medical devices the capability to make decisions and modify output parameters it first needs to be provided with information about the patient’s current status. This is given through vital signals which are obtained through various sensors such as the ones previously discussed in chapter II section 4.

Interfaces Every sensor will use specific communication interfaces to transmit their data depending on the sensors used and the information captured. Examples of such commu-

nication interfaces are analog signals for continuous data generation, standard communication protocols such as UART, CAN, USB, Ethernet, or wireless transmission protocols such as Bluetooth, Zigbee or Ant.

Input device drivers The information provided through the different interfaces is converted through specific device drivers into a digital format that may be used by the application. The sensor manufacturer should provide information for the development of the driver. For analog signals ADC converted are used. Depending on the obtained analog signal characteristics such as resolution, sampling rate and accuracy are selected for the ADC. Once the data is digitalized the acquired adc value is converted into the corresponding real data such as voltage (V), pressure (mmHg) or flow (L/min). A calibration system should be provided to adjust the conversion factors. For digital signals a communication protocol defines also by the manufacturer will define how to obtain the desired values from the transmitted data.

Signal processing Once the data is available from the specific device driver signal processing may be applied. This may include filtering for noise reduction in continuous signals and characteristic extraction to obtain more specific values. This may include mean values, maximum, minimum or QRS extraction for an electrocardiogram signal.

System intelligence After the required signal processing the obtained data may be introduced into the component containing the intelligence of the system which may consist of different controllers. Control based on fuzzy logic with an adaptive mechanism was introduced in the previous section (1). This was used as the main type of control for the system. This allows a straight forward implementation of how to treat the input variables, create rules using these inputs and generate outputs. However future implementations may include other types of intelligent systems such as neural networks.

Output device drivers The output of the intelligent system must be again transformed into signals that may be sent to specific devices. For this again drivers are needed such as digital to analog converters (DAC) or using the previously mentioned communication protocols for input (UART,CAN,USB,Ethernet,ect.)

Actuators The actuators are in charge of converting the received signals generated by the intelligent system into actions. Such devices may be centrifugal blood pumps from an ECSS, where the pump speed may adjusted, medication perfussors that may change the amount and rate at which a medication is administered, a gas blender controlling the mix between air and oxygen.

Supervisor It is critical for a medical device to have a supervisor that is in charge of checking that the system is working correctly. This supervision should be through out all the components of the system, from device drivers, signal processing and decision making. If a failure occurs it should be able to recognize where is the failure, know how critical it is, suggest possible corrections and inform the user, this should be in a graphical manner, sound alarm or an alert light.

Graphical user interface The system needs graphical interfaces for different purposes, these include:

- Provide information about the system.
- Adjust system parameters.
- Interface for calibration of sensor data.
- Allow monitoring of the patient.
- Interface to provide patient information.

Remote Interfaces A remote interface gives the operator the possibility to access the device and know information about the system without having to be close to the device. System upgrades may take place as well as to allow remote monitoring allowing experts to provide advice on how to proceed on specific situations.

3.2. System Requirements

Besides the previously mentioned components required for an automation system, the following are the requirements that were considered:

Robust The application needs to be robust in the sense that it should not contain errors and operate under normal circumstances as well as under unusual conditions without failure. When programs become large and complicated it is difficult to discover errors. It should be possible to easily analyze every part of the system to verify its correct operation.

Reliable It should be possible to rely on the system at all times. Under different conditions it should always do what its expected.

Easily configurable It should be possible to easily create the system and modify it for possible improvements or error correction. If a part of the system needs to be changed, this should not affect the other components.

Energy efficient and low processing The system should not use unnecessary resources, using efficient coding strategies focused on embedded implementations that are of low power and limited processing power. This allows the system to fit in a small form factor and be easily transportable.

Portable It should be possible to port the system into different platforms and operating systems, depending on the needs of the application allowing also the possibility to work with different vendors.

Easy to extend The system should have the capability of easily extending its functionality and integrate new features into the system.

3.3. Framework Implementation

C++ was used as a development language since most of the devices can be programmed with this language. Qt[113] was used for the creation of graphical user interfaces. Since Qt is a cross-platform framework the system to be easily used in numerous operating systems such as Windows, Linux and MacOS. Additionally the Qt source code may be obtained to port the system to embedded x86 or ARM platforms with limited hardware resources. The AutoMedic libraries [17] previously used during experimentation were used as a base for this framework.

3.4. Component Based Development

Considering the previous system description and requirements the implemented framework was built using the concepts of component based software engineering (CBSE). This consists of creating different components with specific functions which can be developed independently, verified and tested. These components are then connected together for the required functionality of a specific application. This allows systems to be modular and be constructed independently.

Under this notion a basic component was defined called *BaseObject*, all of the components of this framework derive from this component. Since the beginning of the development of each component a clear division was done between the implementation of the main functionality that each object should provide and its graphical representation. The *BaseObject* contains the necessary functionality to work without the need of a GUI. A *BaseObjectView* class was designed to contain the graphical representation of each object allowing an easy visualization of the status of each object and their interaction with other components.

BaseObject

The *BaseObject* consists of input and output ports that enable the communication to other components. The *OutPorts* of an object are identified by a unique name. The *ConnectPorts* contain the name of the *OutPort* to which they are referring. It is implemented as a direct pointer to the corresponding *OutPort*.

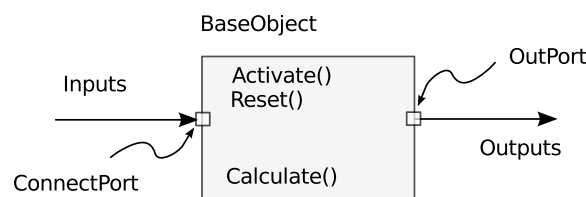


Figure IV-12.: Base Object.

Within the *BaseObject* a virtual *calculate()* function is defined which is implemented by the newly created object. A *Reset()* virtual function is defined to contain the information needed to re-start the specific component and a *Activate()* is used when the device is ac-

tivated. Additional complementary ports (*resetPort*, *activatePort*) are used to control this components through other components.

TriggerObject

Objects that are derived from this type of object have the property of generating events that should trigger the *calculate()* function of all of the objects that are associated to it. An example of this type of object is a timer. This can be configured to trigger an event with a specified period.

BaseObjectView

This class contains the base functions for the graphical representation of an object. A virtual *draw()* function is implemented within the object which is the default graphical representation of all of the created objects, this consists of displaying all of the inputs and outputs, when an input was not connected and if the object is active or not. Derived objects may define their own *draw()* function. Additional mouse functions are included which allow the objects to be controlled through the GUI. All of the derived objects may get mouse action through the *mouseEvent()* virtual function.

The functions used in this class are not dependant on a specific GUI library, a *DrawContext* interface was created to provide the required functions to draw the object such as *drawRectangle*, *drawText*. This *DrawContext* interface may be implemented for a certain GUI library, such as Qt or WxWidgets.

GroupObject and GroupObjectView

A *GroupObject* was created to contain several objects. Its graphical representation is *GroupObjectView* which creates extra panels to show the contained objects in the group.

QtObjectView

An object that extends from this class will provide additional functionalities that are only possible using the *Qt* libraries. This include the creation of user panels using *QWidgets* and the *signal* and *slot* mechanisms provided by Qt. Additionally these type of objects may add panels to the application for further control of its own parameters.

3.4.a. Object Configuration

Configuration files were created to contain parameters that define the characteristics of each object, such as object name, type, input and output port names. For the creation of these files the XML markup language was used, this allows the easy creation of files that can be parsed by the application. The simplest representation of an object is shown in figure IV-13. This code defines a *Sum* object which is named "MySum". It contains two inputs (A and B). These inputs are from another object or objects. The output of this object is called "Result". If the object was programmed as how the name implies, the sum of its inputs, when the *calculate()* function is called and both inputs are connected it will put in the OutputPort of name "Result" the value of $A + B$.


```
<Sum Name="MySum" Output="Result" >
  <Input Name="A" />
  <Input Name="B" />
</Sum>
```

Figure IV-13.: Sum object: basic parameters

The previous object can be easily extended to include other parameters such as parameters for graphical representation as shown in figure IV-14. This new object definition includes parameters that define position and size of the object and the icon to use for its visualization.

```
<Sum Name="MySum" Output="Result"
      X="60" Y="10" W="70" H="80" Icon="Sum.png">
  <Input Name="A" />
  <Input Name="B" />
</Sum>
```

Figure IV-14.: Sum object with view parameters

When the configuration files are parsed the information of the object is gathered in a type of XML node called *AMConfigNode*. Figure IV-15 shows the graphical representation of the previously defined object with two inputs and one output. In this example port "B" is in red showing that it has not been connected. The object is drawn in gray showing that it is not active due to missing input connections.

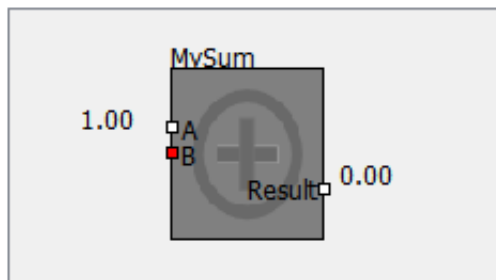


Figure IV-15.: Graphical representation of Sum object.

3.4.b. Object Libraries

The developed objects are contained in different libraries which are loaded dynamically when required. A library called *AutoMedic_base* was created containing the definition of

the previously mentioned objects and functions for loading and parsing the XML configuration files. In order for the objects to be loaded the libraries must implement the function *bool getObjects(XMLNode,GroupObject)*. This is further explained in section 3.4.c. Figure IV-16 displays the different libraries with their containing objects. The OBJ library contains general purpose objects that may be needed in an application. The NFC library contains the objects of a fuzzy controller, an adaptive fuzzy controller and signal filters. It also contains a fuzzy control visualizer to modify the configuration of the controller. The Qt/GUI library contains objects that depend on the Qt libraries, these are visualization scopes and network connectivity. The CHAN library provides a channel class that may store data dynamically and contains frequency information for storing sensor data with different sample rates. The DAQ library contains interfaces that are used by ADC and DAC devices. More detailed description is given in appendix section B.

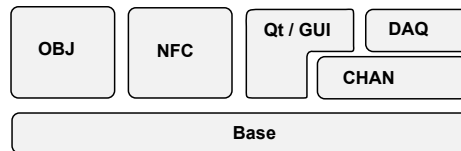


Figure IV-16.: Object Libraries

Device Objects Libraries containing device specific objects can be implemented using the previous libraries. Figure IV-17 shows two basic device implementations.

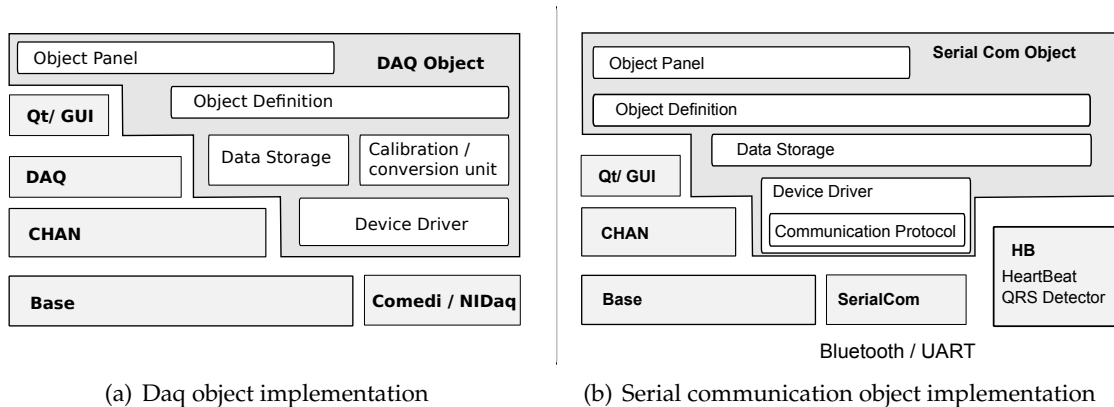


Figure IV-17.: Device specific object library

The figure to the left refers to a data acquisition device. Low level instructions, operating system dependent, are contained in a device driver. This driver may use additional libraries provided by the device manufacturer such as the NI-Daq driver from National Instruments [115] running in Microsoft Windows, or the Comedi [112] driver for Linux supporting a wide variety of DAQs from different manufacturers. A conversion unit is

used to change the digital samples in adc units to conventional units. A calibration system is also provided for correct conversion. The obtained samples can then be stored in data channels. The object definition of these devices is provided by the DAQ library and can be extended to additional functionalities of particular devices. An object panel provides access to the calibration and visualization of data. The implementation to the right refers to devices that use conventional serial communication. This may come from a conventional serial port, a usb or bluetooth with a serial port profile (SPP). A driver containing the communication protocol for each individual device is needed. This must be provided by the manufacturer. Additional libraries such as a heart beat detector can be used if for example the device receives an ECG signal.

For additional information of the different libraries and implemented objects and device drivers refer to appendix B.

3.4.c. Object Server

Once the required components for a specific application have been created and integrated into a library an application is needed to load the objects used by the application and make the proper connections. This application is called an object server.

Application Configuration The object server first of all loads an XML file called App.xml (Figure 3.4.c. This file contains the application configuration: The libraries listed in this file are the ones that are used by the particular application and that will contain the objects used. These libraries are loaded into memory when the program is started. Afterwards the object files listed in the application file are loaded.

```
<Application Name="Device_Control">
  <MDC>
    <LoadConfig Path="Config" W="575" H="325" FullScreen="false" />
    <Objects File="Objects1.xml" />
    <Objects File="Objects2.xml" />
    <Lib File="AutoMedic_OBJ" />
    <Lib File="AutoMedic_QT" />
  </MDC>
</Application>
```

Figure IV-18.: App.xml

The first file contains a Timer object with a period of 100 ms. A group object contains the Constant B with an output value of 10. Additionally a Scope object is added to show the results. The second file contains a sum object called *MySum* which adds the value A and C and puts the value in the output named *Result*. This file also contains a group object called *Inputs* since this group is the same as in the previous file the objects contained are added to the pre-existing group. This consist of another constant and an accumulator that will increase an internal value with the value given by the constant A and will give the result in C .

```
<MDC>
  <Timer Name="t" X="6" Y="49" TimeMS="100" W="59" H="56" />
  <GroupObject Name="Inputs" X="111" Y="53" W="66" H="80" >
    <Constant Name="B" Value="10" X="42" Y="94" W="50" H="56" />
  </GroupObject>
  <Scope Name="Scope" NumScopes="1" X="379" Y="55" >
    <SubScope Id="0" Name="SumScope" NumChannels="4" AutoScale="true">
      <Chan Id="0" Name="A" Color="black" Units="_" Freq="1" />
      <Chan Id="1" Name="B" Color="red" Units="_" Freq="1" />
      <Chan Id="2" Name="C" Color="green" Units="_" Freq="1" />
      <Chan Id="3" Name="Result" Color="blue" Units="_" Freq="1" />
    </SubScope>
  </Scope>
</MDC>
```

Figure IV-19.: Objects1.xml

```
<MDC>
  <Sum Name="MySum" Output="Result" X="235" Y="56">
    <Input Name="A" />
    <Input Name="C" />
  </Sum>
  <GroupObject Name="Inputs" >
    <Constant Name="A" Value="0.01" X="41" Y="18" W="50" H="56" />
    <Accumulator Name="Accumulator" Input="A" Output="C" X="146" Y="15" />
  </GroupObject>
</MDC>
```

Figure IV-20.: Objects2.xml

Object Loading The ObjectServer application contains a *GroupObject* that will contain all the objects of the application. This *GroupObject* is extended to a graphical representation (*GroupObjectView*) which handles the mouse actions and the drawing of all the objects contained in the group.

In the previous example the files called *Objects1.xml* and *Objects2.xml* are loaded. The application will read the first file and will go through each object, represented as an XMLNode. In this case `<Timer ... />` will be the first XMLNode. This node is sent to each library by calling the *bool getObjects(XMLNode,GroupObject)* implemented in each library as previously described in section 3.4.b. When a specific library finds that it contains the implementation of the object called *Timer* it will create a new instance of this object sending the XMLNode to the object constructor for extract of its parameters. This new object instance is then added to the *GroupObject* sent by the ObjectServer application. The function will then return with a true value making the ObjectServer look for the next object. If on the other hand the object was not contained in this specific library it will continue searching in the rest of the libraries until it returns true. If the object was not found in any libraries the ObjectServer will show an error and will not continue.

Object Connection Once all of the objects are loaded the Object Server makes all of the corresponding connections using the input and output names and assigning the pointers

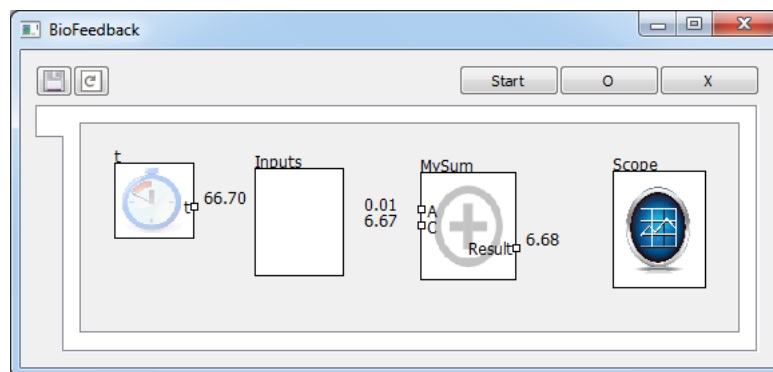
on the ConnectPorts for each input.

Calculation Trees After these connections are done the server object creates a calculation tree based on input and output dependencies starting from the trigger object. In this case the calculate tree shown in figure 3.4.c is created. Currently the created trees are unary, meaning the operations are sequential, however with multiprocessor and multithreaded systems it may be possible to create more complex trees with parallel processing for non dependent objects.

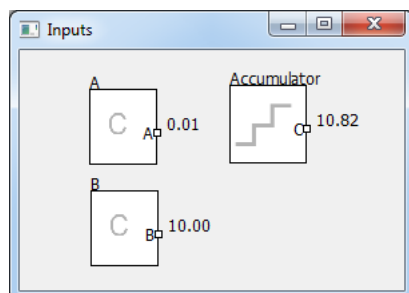


Figure IV-21.: Calculate tree of program

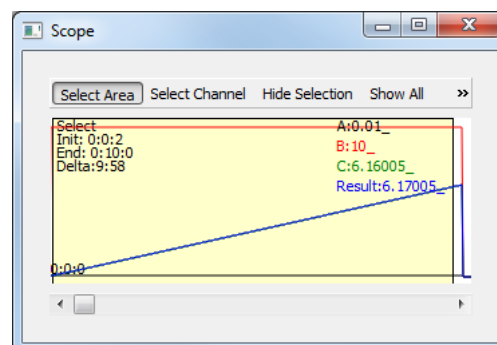
Figure 3.4.c shows the graphical representation of the previous application once the objects have been connected and the calculate tree created. Figure IV.22(a) shows the main window of the Object Server. Figure IV.22(b) shows the objects contained in the GroupObject Inputs and IV.22(c) is the window of the scope.



(a) Application main window



(b) Inputs Group



(c) Scope Object

3.4.d. Advantages of using the AutoMedic Framework

The described framework clearly brings several advantages over a custom made program. These are some of the most relevant.

- Objects may be developed and analyzed independently; this allows developers to work in parallel focusing in specific functionalities of the system. Tests may be designed to assure the correct operation of the object and then they can be easily assembled with the complete of the application.
- New hardware or changing a specific sensor may be easily done simply by replacing the specific library and providing the same outputs as the previous one.
- The step from testing the system through simulation to using the system with real sensors is simple by just introducing the sensor objects instead of the ones that are used provide the simulation data.
- Using a component based approach allows the same objects to be used in different applications, decreasing the amount of time needed for development and making the error findings faster and easily corrected in all of the applications.
- With all the libraries created and compiled, application specific configurations may be easily created without the need of recompilation of the code.
- If a new component is found to have an error, only the specific module needs to be changed without putting at risk the rest of the system.

3.5. Embedded System Implementation

An already built embedded system was chosen to port the developed framework and test its capabilities. Several options were found as evaluation boards [116–118]. From Technologic Systems [119] an embedded system was chosen since it covered all of the previous requirements.

3.5.a. Hardware specification

The embedded system consists of a TS-7300 board (figure IV-22 with an 200Mhz ARM920T processor, 2 Ethernet ports, 2 USB, 2 UARTs, 2 SD cards, along with a PC/104 Bus for additional daughter boards. The TS-ADC16 board was connected to the PC/104 bus, with 16 ADC channels with 16-bit resolution and four 12-bit DAC channels. An additional daughter board was added to provide CAN bus connectivity. The system consumes 2.8 watts making it a very low power device that may be easily used with batteries.

3.5.b. Software specification

The TS-7300 board comes with a pre-configured open source Linux Debian based distribution with Kernel version 2.4.26. Having this operating system provides functionality such as Ethernet configuration, ssh remote access, sd read/write access. Embedded Qt

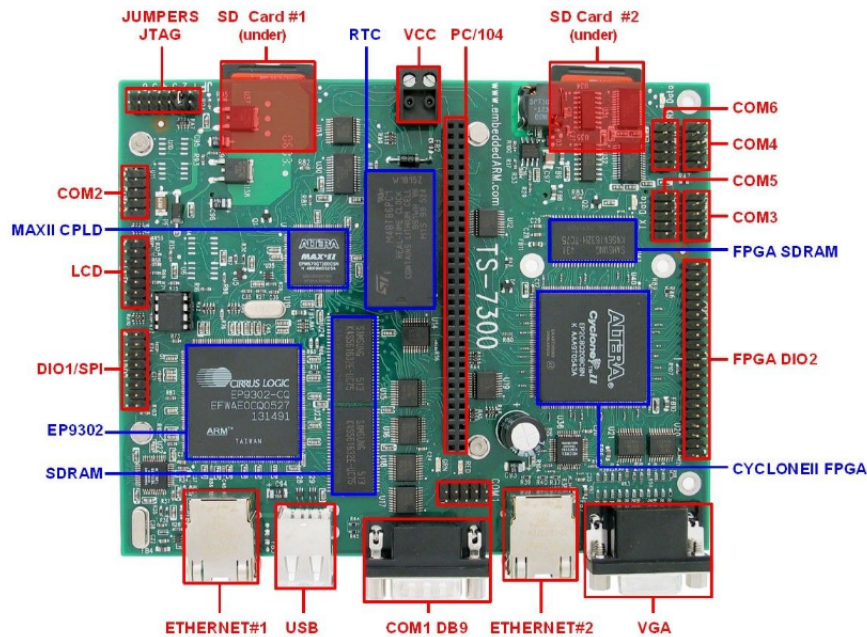


Figure IV-22.: TS-7300 Board

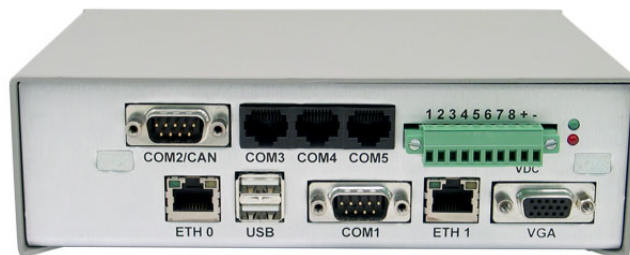


Figure IV-23.: TS-7300 Board Enclosure

version 4.7 [113] was used for this implementation. It provides a lightweight window system consisting of a direct connection to the device framebuffer, eliminating the need for the X Window System saving memory. The Qt source code downloaded and with a gcc cross-compiler provided by Technologic Systems Qt was compiled.

Figure IV-24 shows the different components of the software layer. With the possibility of using Qt in the embedded system all of the libraries previously mentioned may be compiled to the arm architecture. Only specific components such as the ADC may be added with additional libraries. In the application configuration this only requires the change of a component type, without affecting the rest of the components. Additionally once the libraries are compiled together with the object server, if the controller needs to be changed or any of the components, the whole system does not need to be recompiled, requiring changes only from the xml configuration files.

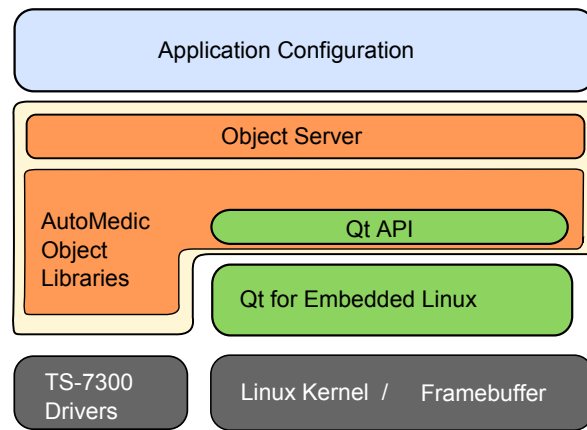


Figure IV-24.: Software Layer

3.5.c. Supervisor System

To verify the correct operation of the system several supervising components were considered as shown in figure IV-25.

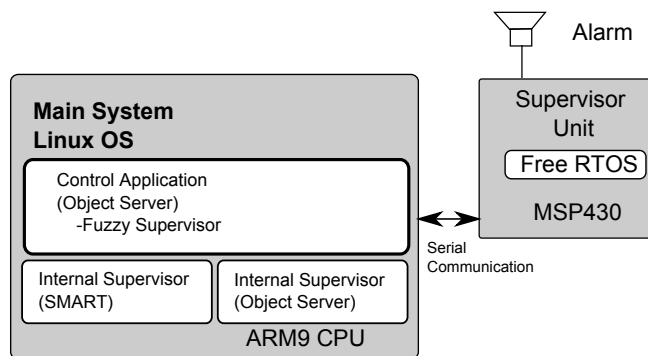


Figure IV-25.: Supervisor system

The first supervising component is part of the control application, implemented with an ObjectServer containing a Supervising Fuzzy Control previously explained in this chapter in section IV. This supervising component is in charge of checking the status of the patient, and that all the sensor data is correct. The second supervising component is based on a software monitoring system that runs in parallel with the control application. This may be an additional ObjectServer receiving constant information from the control application or another application such as SMART (Smart Monitoring and Rebooting Tool). SMART is an open-source program that runs as a daemon in the Linux operating system. In case a service is not present or stops working SMART automatically restarts the process. The third supervising component consists of an additional micro processor capable of running independently of the main system. In the system implemented both systems communicate periodically through serial communication sending each other an alive signal and

acknowledgment. A state machine was created to allow both of the systems to communicate with each other. If one of the systems stops responding an alarm is generated to inform the user of a system failure.

For the external supervising unit the FreeRTOS real time operating system was used together with a small application which was programmed into a TI MSP430 low power 16-bit microcontroller[120]. The FreeRTOS provided the functionality needed and was easily ported to the selected hardware.

4. Portable ECSS Control

This section uses the previously described software framework to implement the automation system of the ECSS. This also takes into account what was learned experimentally from chapter III and the perfusion considerations from section II.3.

The automation of the ECSS was divided into four main controllers, one in charge of adjusting the centrifugal pump speed, a second one adjusting the FiO₂, the third for the Gas Flow and a fourth one for the administration of medication. These controllers are able to work independently, however the interaction between the parameters to control were also taken into consideration. For each controller a knowledge base was generated with smaller sub-controllers describing the desired behavior depending on specific inputs. An adaptive controller was used integrating all the inputs from the knowledge base, with the purpose of generating new rules based on what is suggested from the knowledge base controllers and how the patient responds to the given perfusion.

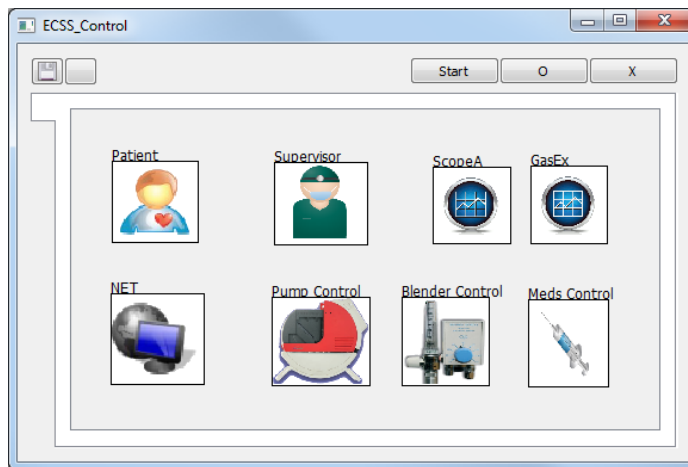


Figure IV-26.: Main window of ECSS control system

4.1. Parameters

Table IV.4 contains the parameters that were taken into consideration. These parameters may be acquired during ECC with different sensors and devices. For each signal a "critical minimum" and "critical maximum" is configured. The input signals should not be outside this range otherwise an alarm is triggered. A minimum and maximum value is a range in which the signal may be without representing a high risk to the patient. A normal value is the value set to the controller as target for each signal. Additionally a weight may be established for each parameter, where the operator may define the importance for each parameter. The second part of the table shows the output variables, specifying the minimum and maximum permitted values together with the initial value at which the output starts when the controller is initialized.

Input Variables						
Name	Critical Min	Min	Normal	Max	Critical Max	Units
MAP	40	50	70	110	140	<i>mmHg</i>
EFR	0	1	4	5	6	<i>L/min</i>
SVR	400	500	736	1600	1900	<i>dyn · · · /cm⁵</i>
SpO _{2a}	60	70	99	100	101	%
SpO _{2v}	40	50	76	80	90	%
PO _{2a}	70	90	100	110	140	<i>mmHg</i>
PO _{2v}	25	30	40	60	80	<i>mmHg</i>
PCO _{2a}	20	30	40	50	60	<i>mmHg</i>
PCO _{2v}	25	30	46	50	60	<i>mmHg</i>
Control Variables						
PumpSpeed	-	1000	1000	3900	-	<i>rpm</i>
FiO ₂	-	20	80	100	-	%
Gas Vol	-	2	5	10	-	<i>L/min</i>
ISDN	-	0	0	2	-	<i>ug/kg/min</i>
NEP	-	0	0	0.1	-	<i>ug/kg/min</i>

Table IV.4.: Control Parameters

4.2. Centrifugal Pump Control

The centrifugal pump is the main component of the ECC in charge of perfusion. It affects directly the blood flow and pressure and as a consequence is a major factor in changing oxygen delivery and carbon dioxide removal. The pump speed is changed by setting the desired rounds per minute (rpm) which may be increased to a maximum of 3900 rpm and decreased to a minimum of 1000 rpm to prevent any back flow in case the patient is at a higher position than the ECSS. Below this speed the arterial and venous clamps should be closed.

The reference models used for the EFR and MAP were adjusted to have a settling time of 30 seconds.

4.2.a. Knowledge Base

The designed pump controller uses as knowledge base four controllers. The first and the second are in charge of monitoring the MAP and the EFR respectively. The two other controllers are in charge of monitoring the gas exchange, one monitors the FiO₂ and gas flow and the other the oxygen delivery and consumption.

MAP and EFR Control Both of the MAP and EFR controllers use as inputs its reference signals and the change over time to modify its pump speed depending on the reference model. Their rule base is based on a Fuzzy-PI controller. A characteristic observed in the reference model control is that if the plant, in this case the patient, is reaching the target of a specific signal faster than what is predicted by the model, the controller will try to slow

down the system to resemble the model causing some possible oscillations. In the case of the controllers used for the ECSS, if the patient is reaching its target of pressure or flow faster than what is predicted, it will not try to slow down the system but will wait to take action until the reference model is the same as the patients signal. This way the signals are expected to react not slower than the reference model, and in case its faster, the controller will not make any adjustments until its necessary. To introduce this into the controller in the form of rules an additional signal was introduced referring to the target error. This will describe four cases and are shown in figure IV-27:

1. If the reference error and the target error are both positive or both negative (A,B) then the controller should take action and modify the output according to the normal Fuzzy-PI rules.
2. If the reference error has a different sign than the target error then the signal is reacting faster than the reference model is reaching its target sooner. For this case the controller should not take action. (C,D)

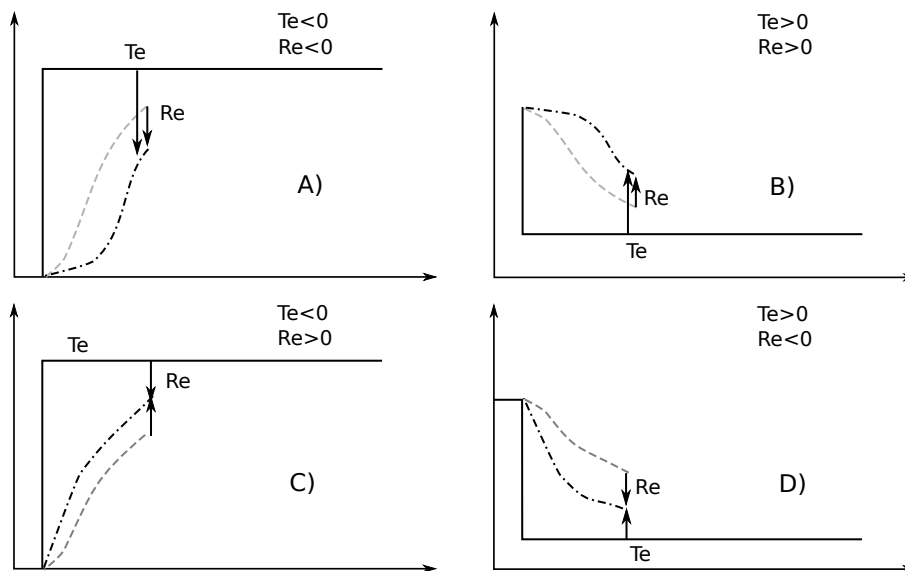


Figure IV-27.: Four types of error with respect to reference model.

Gas Exchange One controller is in charge of checking if the flow generated by the pump is sufficient for the desired venous oxygen saturation. If both FiO_2 and gas flow are set at its maximum and the venous oxygen saturation is still low, this controller will try to increase the pump speed. The same will happen if the gas flow is at its maximum and the venous PCO_2 is still high. The last controller defines the rules depending on the oxygen delivery and oxygen consumption. If both are lower than the expected target then they will try to increase the pump speed.

The output of each controller is multiplied by a weight to set the level of importance for each signal.

4.2.b. Adaptive Controller

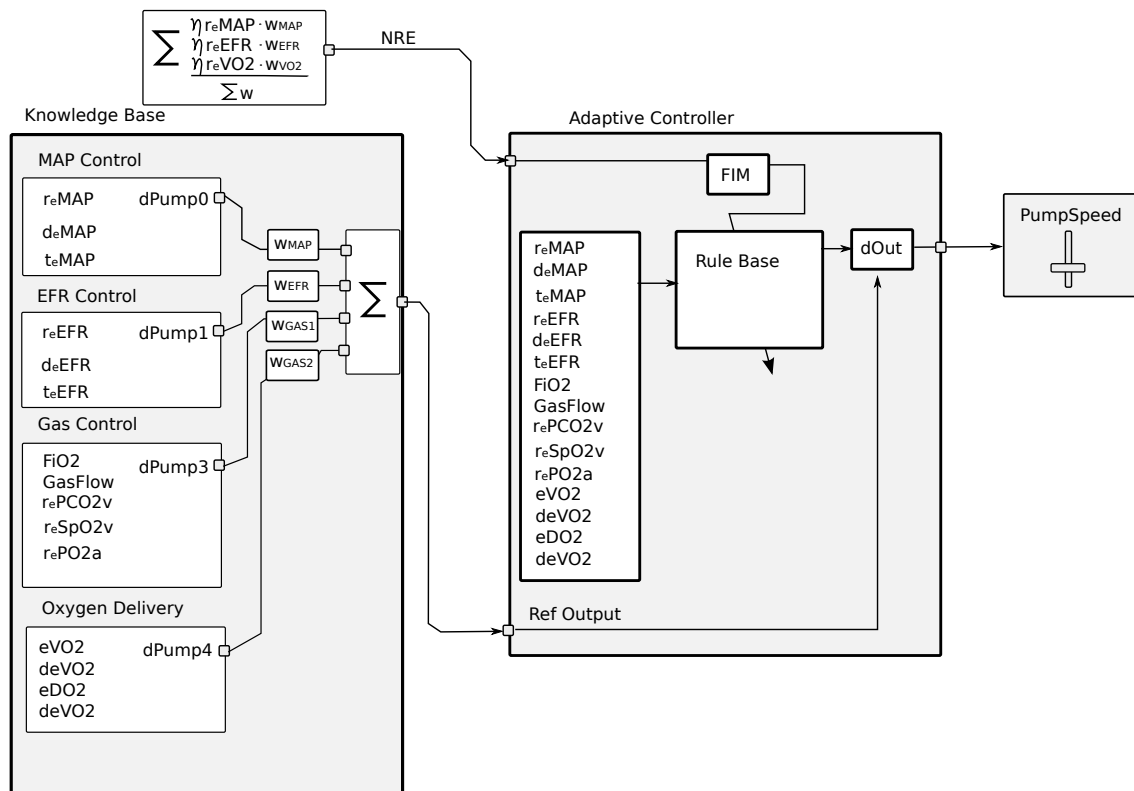


Figure IV-28.: Pump Control with Knowledge Base controllers and Adaptive Controller

An adaptive controller was introduced into the pump controller as shown in figure IV-28. The reference output signal is taken from the output sum of the previously defined knowledge base controllers. The same input signals are used as the previous controllers, however the number of sets used for each variable was increased. The granularity of the output variable was set to 31 sets distributed in a focused manner towards the middle set. For the correction of the rule base the NRE was calculated based on the normalized reference error of MAP, EFR and VO₂ and its corresponding weights.

4.3. Gas Blender

The controller for the gas blender monitors the oxygen and carbon dioxide in the arterial and venous line. This data is provided by the gas analyzer. The gas blender provides the oxygenator with a gas mix of normal air and compressed oxygen. In the gas blender two parameters may be adjusted: the fraction of inspired oxygen (FiO₂), which modifies the gas phase oxygen tension, and the amount of gas flow. Previous research of blood-gas control include [88, 91, 110] As general practice for perfusion FiO₂ is used to regulate the oxygen saturation, while gas flow is used to control the carbon dioxide removal, however they both relate with each other; if FiO₂ is at its maximum and the oxygen target values are still not reached gas flow may be increased to improve oxygen delivery.

4.3.a. FiO2 Controller

The FiO2 allows the regulation of oxygen saturation in the blood. The reference models for partial pressures and saturation were configured with a settling time of 20 seconds. If the venous oxygen saturation is low immediate response is needed from the FiO2 controller.

The general rules used for this controller are the following:

1. If SpO2a is low or the SpO2v is low then increase FiO2.
2. If SpO2v is high then decrease FiO2.
3. If PO2v is normal and PO2a is high then decrease FiO2.

These rules were integrated into 3 sub controllers to form the knowledge base. These controllers dictate the increase or decrease of the FiO2. One monitors the venous SpO2, a second the arterial PO2. These controllers implement a Fuzzy-PI rule base, however similar to the MAP and EFR controllers the target error is included in case the signals are reaching its target faster than their corresponding reference model. A third controller gathers the information of the arterial and venous PO2 to decrease the FiO2 in case both values are high, For the other cases this last controller remains inactive.

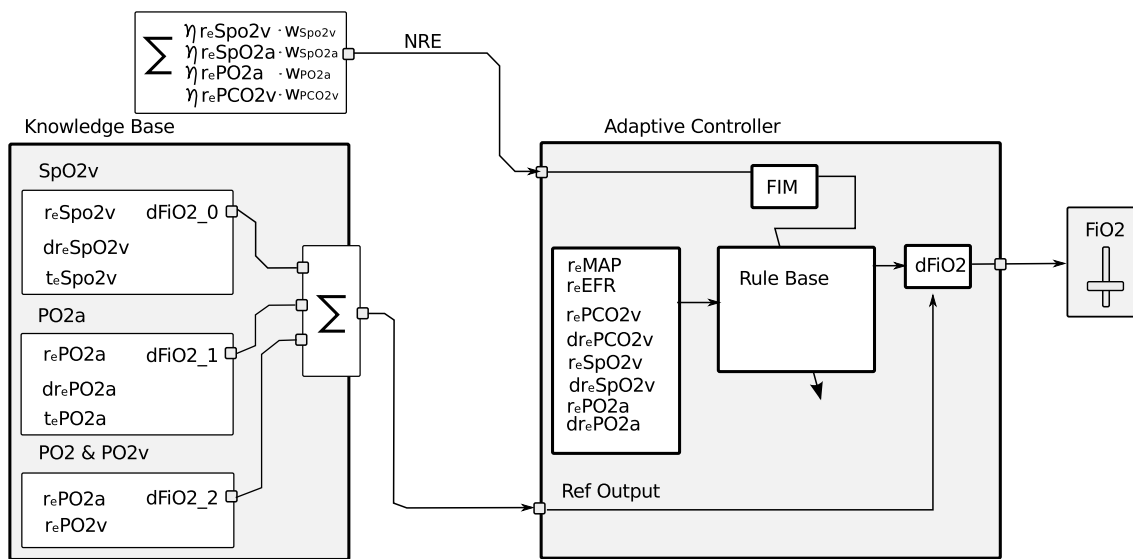


Figure IV-29.: Blender FiO2 Control

4.3.b. Gas Flow

The gas flow controller is mainly used to regulate the carbon dioxide. By increasing the gas flow it is possible to increase the amount of carbon dioxide removed from the blood introduced into the oxygenator. This was expressed in a sub controller monitoring the venous partial pressure of carbon dioxide.

An additional controller was introduced to increase the gas flow in case the FiO2 is at 100% and the oxygen partial pressures in the arterial or venous line have not been reached.

An adaptive controller was included using the same inputs as the knowledge base controllers plus the MAP and EFR reference errors. The NRE used for adaptation was calculated using the reference error values of SpO2v, SpO2a, PO2a and PCO2v with their respective weights.

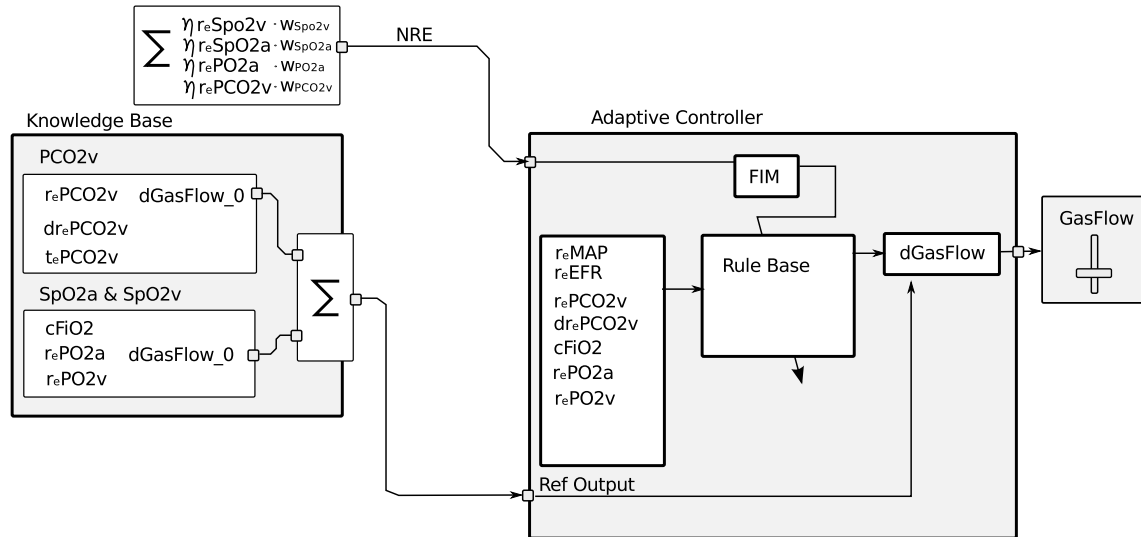


Figure IV-30.: Blender Gas Flow Control

4.4. Medication

The medication controller is focused on the administration of vasoconstrictor or vasodilator substances that may improve organ perfusion. The behavior of this controller is defined based on the target values of MAP and EFR. From these two values an extracorporeal systemic vascular resistance (EccSVR) is calculated based on the following formula:

$$EccSVR = \frac{MAP}{EFR} \quad (IV.10)$$

Since both medications considered have contrary effect, one acting as a vasoconstrictor and the other a vasodilator they were joined into one control variable (Meds). When this variable is positive that means that a vasoconstrictor should be applied; if the variable is negative then a vasodilator should be applied. As knowledge base a Fuzzy-PI type of controller was used to monitor the EccSVR; if the reference EccSVR error is high then the Meds variable is decreased, if its low then the Meds variable is increased. The reference models were set to have a settling time of 5 minutes to give time for the body to respond to the medication.

For the adaptive controller errors of MAP and EFR were included besides the inputs from the knowledge base. The NRE was calculated based exclusively in the EccSVR normalized reference error.

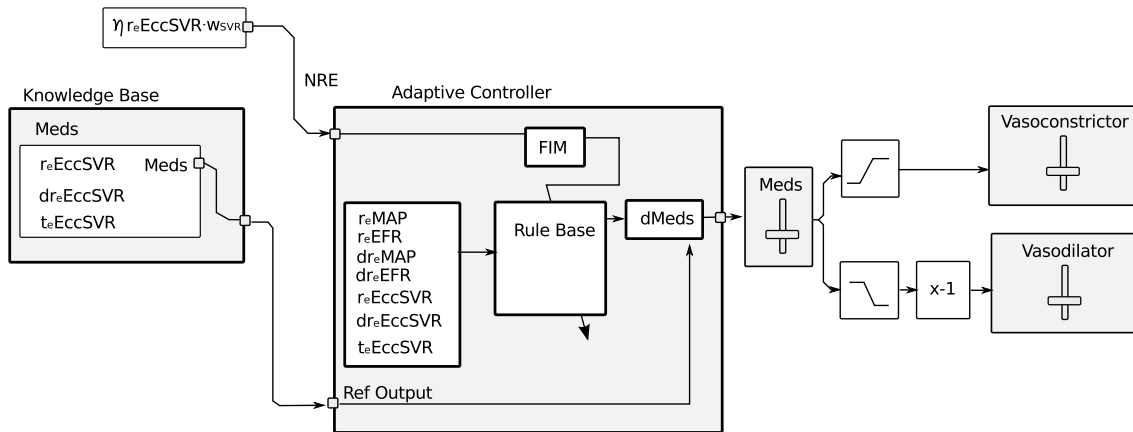


Figure IV-31.: Medication Control

4.5. Implementation

The previous controllers were described in an XML format using different objects according to the specifications of the AutoMedic framework. The framework gathers these objects, checks for any errors and makes all the proper connections and calculation steps. The generated files are shown in the appendix section D.

Control and Simulation Results

Contents

1. Control Learning and Adaptation	109
2. ECSS Control Results	112

This chapter combines the work of the mathematical model and simulation developed in chapter III together with the designed ECSS controller of chapter IV. On the first part the results of several simulations will be showed allowing the evaluation of the different control mechanisms. These simulations will focusing only on the control of the pump speed. In the second section all the designed controllers are connected with the simulation environment and simulated over different test cases.

1. Control Learning and Adaptation

This section evaluates the different mechanisms that were designed for adaptation. To do this a single flow rate controller was used in charge of adjusting the pump speed. Three cases were evaluated, the first results show the controller without previous knowledge and with the adaptive mechanism active. The second results show the controller response when only previous knowledge is used but no adaptive mechanism is active. The last results show the controller with previous knowledge and the adaptive mechanism active. For all the tests a reference model was used to represent the expected behavior of the controller to reach the target EFR.

To compare the different control mechanisms the mean absolute error (MAE) was calculated between the reference model and the current signal with formula IV.9

$$MAE = ABS(Ref - Current) \tag{V.1}$$

The simulation model used was the complete cardiovascular system with an active heart. A target signal was programmed to change between 3 and 5 L/min with a period of 10 seconds.

1.1. Adapting without previous knowledge

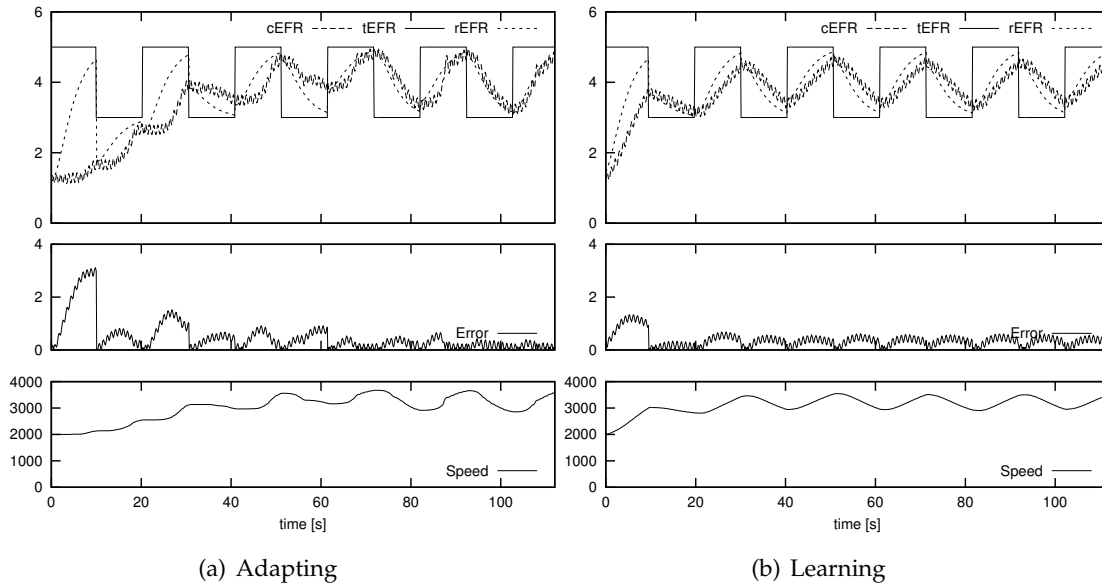


Figure V-1.: Results of control mechanisms

For this test the rule table of the controller started empty. In every step the controller verifies if there is no existing rule with the corresponding active input sets. If a rule does not exist it is created using the active sets and the output set of the rule is set to zero. After a delay of 10 control steps the adaptive algorithm was activated and the controller rules were modified according to the previous adaptive mechanism mentioned in chapter IV. Since the controller had an empty rule base it did not do any correction at the beginning, then the output sets of the corresponding active rules at t -delay with a truth higher than a threshold were slowly modified to make the system reach the reference signal. After 900 seconds the controller was able to follow the reference model. This is shown in figure V.1(a).

1.2. Learning without adaptation

Figure V.1(b) shows the results of the controller with only previous knowledge is active. This is equivalent to having a static controller. The simulation shows that the controller was almost able to follow the reference model since the beginning however further adjustments are needed to further reduce the MSE. Since no adaptive mechanism is active this error always remains the same.

1.3. Learning and adaptation test

Figure V-2 shows the controller with the knowledge controllers provided for learning and the adaptive mechanism active. This solutions shows the best results since the controller is able to follow closer the reference signal since the beginning and do further adjustments to the rule table to reduce to the minimum the MSE.

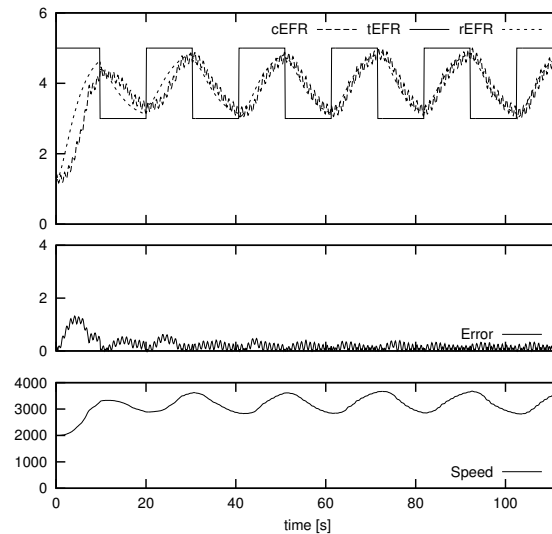


Figure V-2.: Results of learning and adaptive controller.

1.4. Control mechanism comparison

Figure V-3 shows an accumulated MAE (AccMAE) which allows a comparison of the different control mechanisms. An additional result was included referring to the fuzzy controller used as knowledge base. From these four results we can see that the learning part of the algorithm is capable of replicating the behavior of the simple fuzzy controller with a small increase of the mean AccMAE do to the number of output sets used that may generate a round-up. The adaptive mechanism without knowledge has the worst response since it has to create its own rule-base, however once it adapts to the system it may be capable of reducing the rate of increase of the MAE and in a long term simulation even improve the performance compared to the simple fuzzy controller and knowledge based. The result of both the learning and adaptive mechanism (L+A) activated showed the best results almost since the beginning, since it was able to quickly create its own rules using as reference the knowledge base controller and at the same time improve the generated rules with the adaptive mechanism.

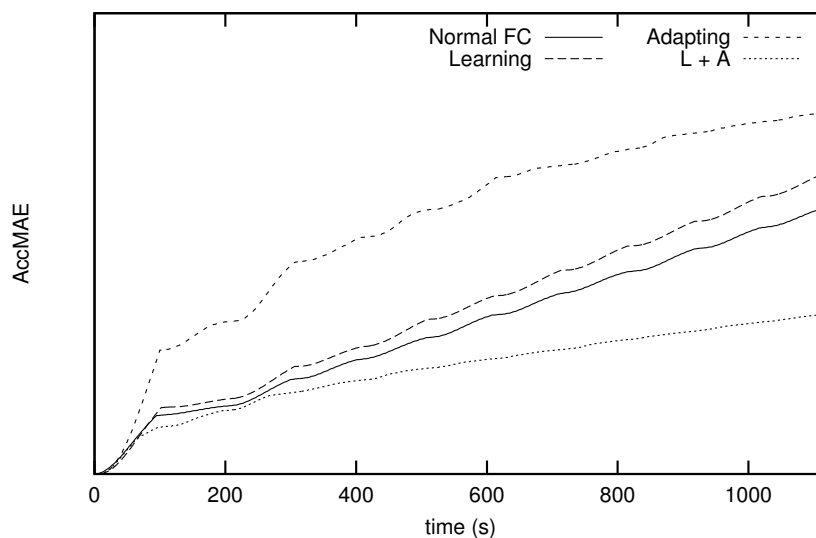


Figure V-3.: Comparison between control mechanisms.

2. ECSS Control Results

This section evaluates the designed controllers for the ECSS using the simulation model. Two cases are represented: the first one considers a patient with beating heart and a high pressure when connected to the ECSS. The second case considers the opposite case where the heart has stopped beating and the patient has low arterial pressure.

For both cases four types of controllers were evaluated through different simulations. The first two controllers consider a normal fuzzy controller using directly the output from the knowledge controls without any modification. The first case leaves out the application of medication (*FC / no Meds*), while in the second the medication controller is activated (*FC / Meds*). The two last controllers consist of an already trained adaptive fuzzy controller, again considering one case without medication (*AFC / no Meds*) and a second with medication (*AFC / Meds*).

2.1. Case of beating heart and high pressure

This simulation represents the case of a patient that still has a beating heart, but due to a disfunction such as a mitral regurgitation caused by an inappropriate closing of a valve, the heart is not able to generate sufficient cardiac output. Additionally high blood pressure is considered for this case.

Model setup

To increase the arterial pressure in the mathematical model the pulmonary resistance was increased together with the resistance in the systemic circulation. The cardiac output was decreased by generating a mitral valve regurgitation creating a backflow from the left ven-

tricle to the left atrium. Additionally the heart elastance was also decreased to further decrease cardiac output.

Control setup

The target pressure was set to 60 mmHg and the flow rate to 5 L/min. From these two target values a SVR of 960 was set as target. This value was obtained from the following formula:

$$EccSVR = 80 \cdot \frac{MAP}{EFR} \quad (V.2)$$

Regarding the control of the gas exchange the expected arterial partial pressure (PO_{2a}) was set to 100mmHg while the oxygen saturation at the venous side (SpO_2) was set to 80%. For the target carbon dioxide partial pressure was set to 42 mmHg.

Control response

Figure V-4 and V-5 shows the response of the four control simulations. In this case target flow rate could not be achieved by increasing the pump speed, due to the presence of a high pressure. The controller in charge of monitoring the pressure tries to decrease the pump speed while the controller in charge of reaching the EFR is doing the opposite. According to the target values of MAP and EFR the desired SVR is calculated and shown in the third graph. This indicates the need of a vasodilator to be able to reach the target values. For both cases of no medication the EFR was limited to only 4L/min and the pressure was higher than 60mmHg. For the cases where medication was applied the systemic vascular resistance could be reduced with the application of ISDN. The medication controller was activated only after the EFR and MAP signals were stabilized. With the simple fuzzy controller a big overshoot was produced in the flow, reaching 6L/min while the pressure decreased to 55mmHg. With the use of the trained and adapting controllers the oscillations were significantly reduced and the pressure and flow could stay closer to their target values. Below the pump speed a graph shows the administration of a vasodilator (NEP). Once the vasodilator takes effect the pump speed is slowly increased by the controller to reach the target values of EFR without exceeding the MAP.

For this example if the medication controller is not activated it is not possible to reach the EFR with the desired target pressure. Still if the application of medication is restricted the operator may modify the weight of either the pressure or flow to give a higher importance to one of the controllers and thus come closer to the target value.

The reference models for both MAP and EFR were faster than the actual response of the cardiovascular system due to the high vascular resistance mentioned previously. These two reference models are also used as a safety measure to reduce oscillations in case the response of the system is faster than what is being calculated by the reference models. The reference model of the SVR was configured to have a slower response than the EFR and MAP. This model was closely followed by the system with the application of medication once the controllers were properly trained.

Figure V-5 presents the results of the previous simulation case focusing on the gas exchange control. The first graph shows the arterial oxygen saturation, once the arterial

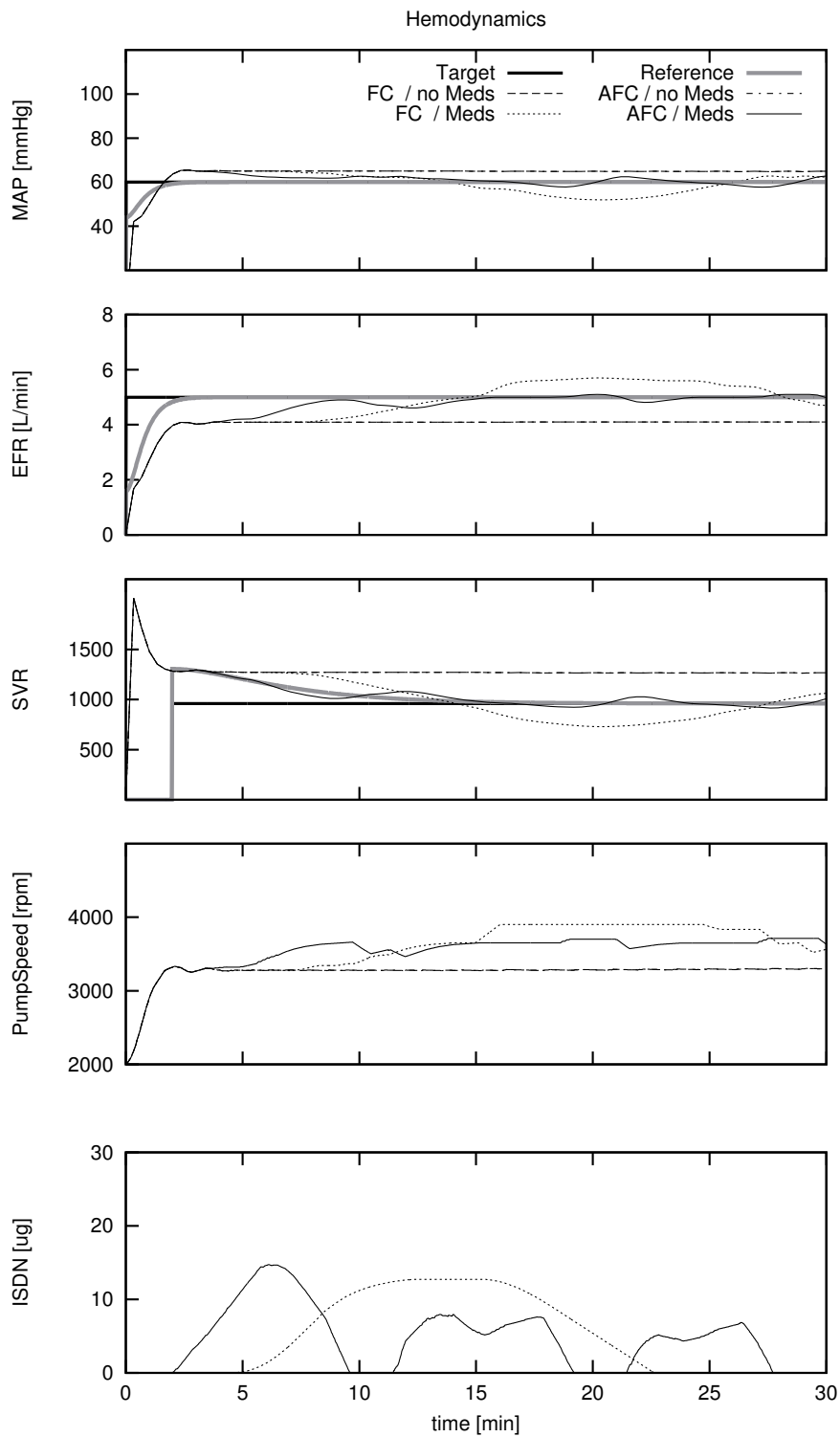


Figure V-4.: Hemodynamic results

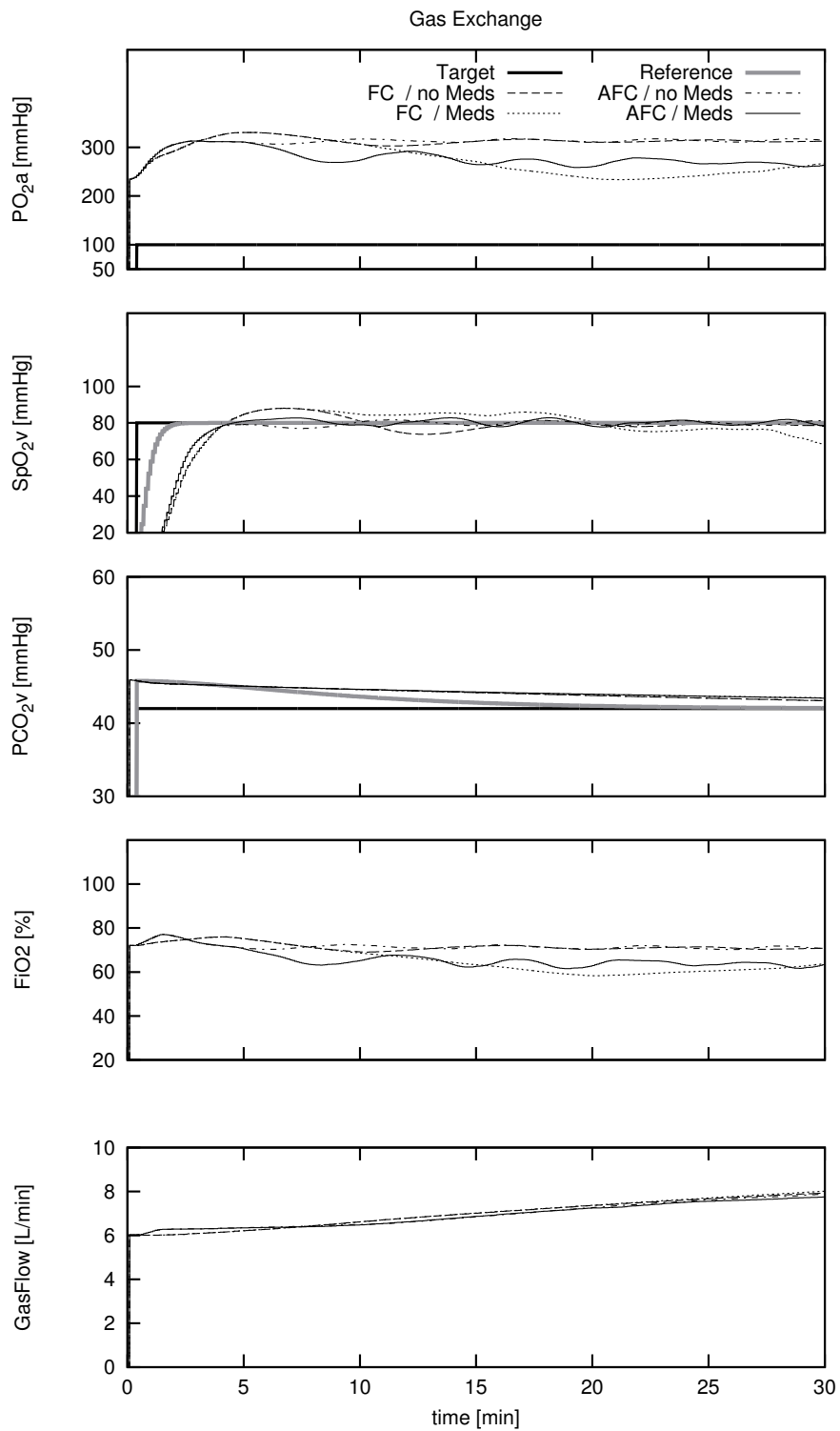


Figure V-5.: Gas exchange results

cannula is opened this pressure is quickly increased. The second graph shows the oxygen saturation at the venous side, once the extracorporeal circulation is initiated and oxygen rich blood is introduced into the body the venous oxygen saturation gradually increases.

In relation to the carbon dioxide, when the ECC was initiated the venous partial pressure of CO_2 was higher than the target. To reach the target value the controller increased the gas flow.

2.2. Case of non beating heart and low pressure

Model setup

These simulation results show the case where the heart is completely stopped and with low arterial pressure. To achieve this the systemic resistance of the cardiovascular model was decreased reducing the over all pressure.

Control setup

In this test case the ECSS must be able to fully take over the heart function. Additionally as seen from the perfusion section (II.3) it is necessary to keep a mean arterial pressure between 50 and 60 mmHg to guarantee organ perfusion. For this case a vasoconstrictor is needed to reach the desired target values of flow and pressure.

Control response

Figure V-6 shows the results of this simulation. Both the MAP and EFR started at a low value and as soon as the machine was connected the pump speed increased starting from 2000 rpm. Due to the low vascular resistance in the system the EFR at first surpassed the target value trying to reach the desired MAP. Increasing the EFR however was not sufficient and the pump speed reached its maximum speed of 3900 rpm. Similar to the previous case for the controllers without medication the EFR and MAP remained constant for the rest of the simulation. For the cases where the medication controller was activated NEP was applied to increase the SVR. This allowed the pressure to increase to 60 mmHg and reduce the EFR to its target. Additionally the pump speed could be reduced eliminating the need to run at full speed. After reaching the desired SVR the overshoot was compensated with a dosage of ISDN.

With respect to gas exchange, the FiO_2 controller was configured to rapidly increase if the venous oxygen saturation was below its target. This was done to provide the patients body with oxygen as soon as the ECSS is connected. Afterwards the oxygen administration is slowly decreased to reduce the arterial partial pressure. An overshoot was produced due to this effect, for the controllers without adaptation (FC) the oxygen saturation reached levels lower than its target and some oscillations were produced when trying to correct the overshoot. For the AFC with medication the rules were adapted so that the oxygen saturation would gradually decrease until reaching the target saturation level without producing significant oscillations or reaching lower saturation levels. The carbon dioxide partial pressure started high, but the gas flow was increased starting from 6 L/min effectively reducing CO_2 levels.

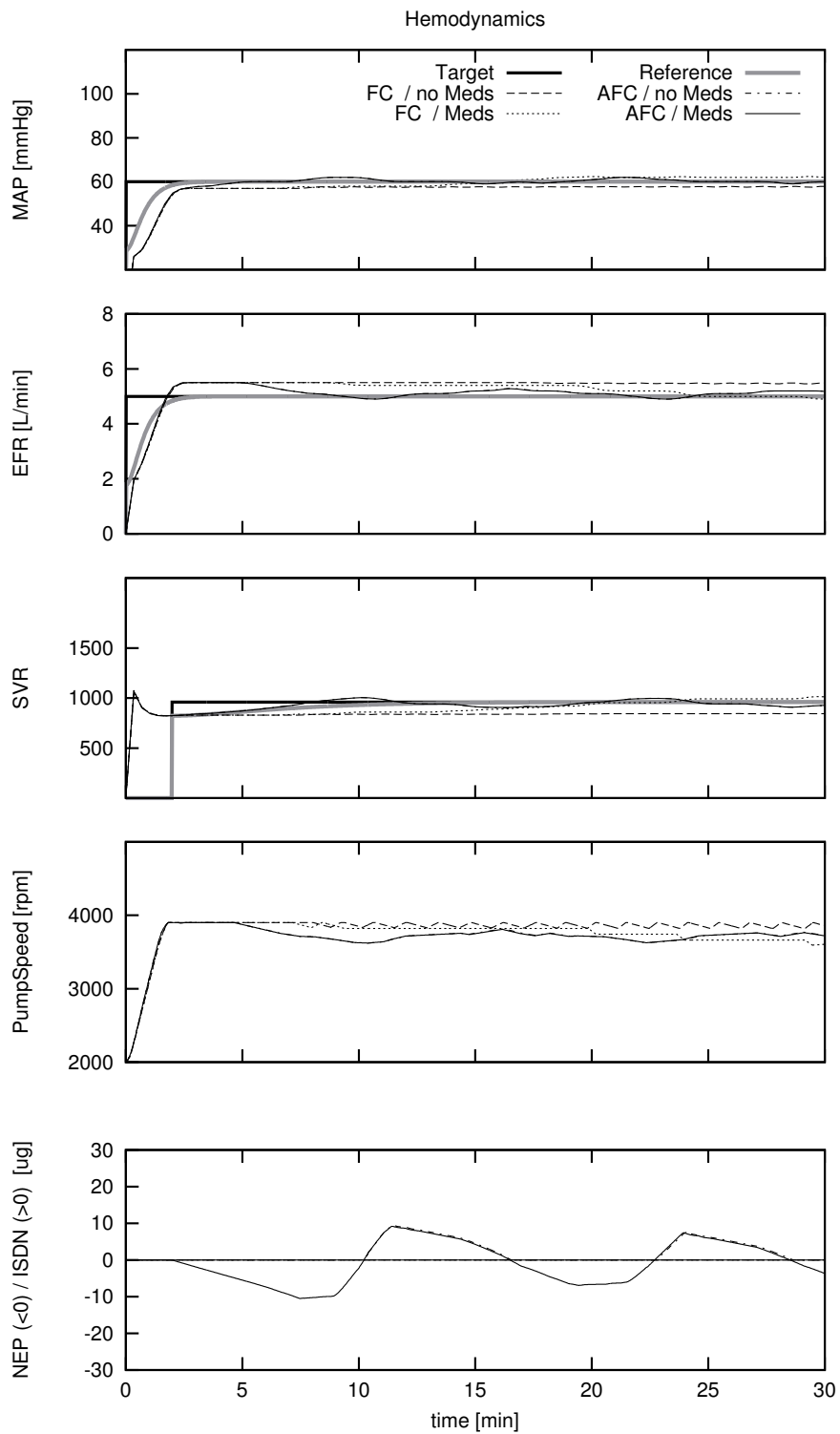


Figure V-6.: Hemodynamic results

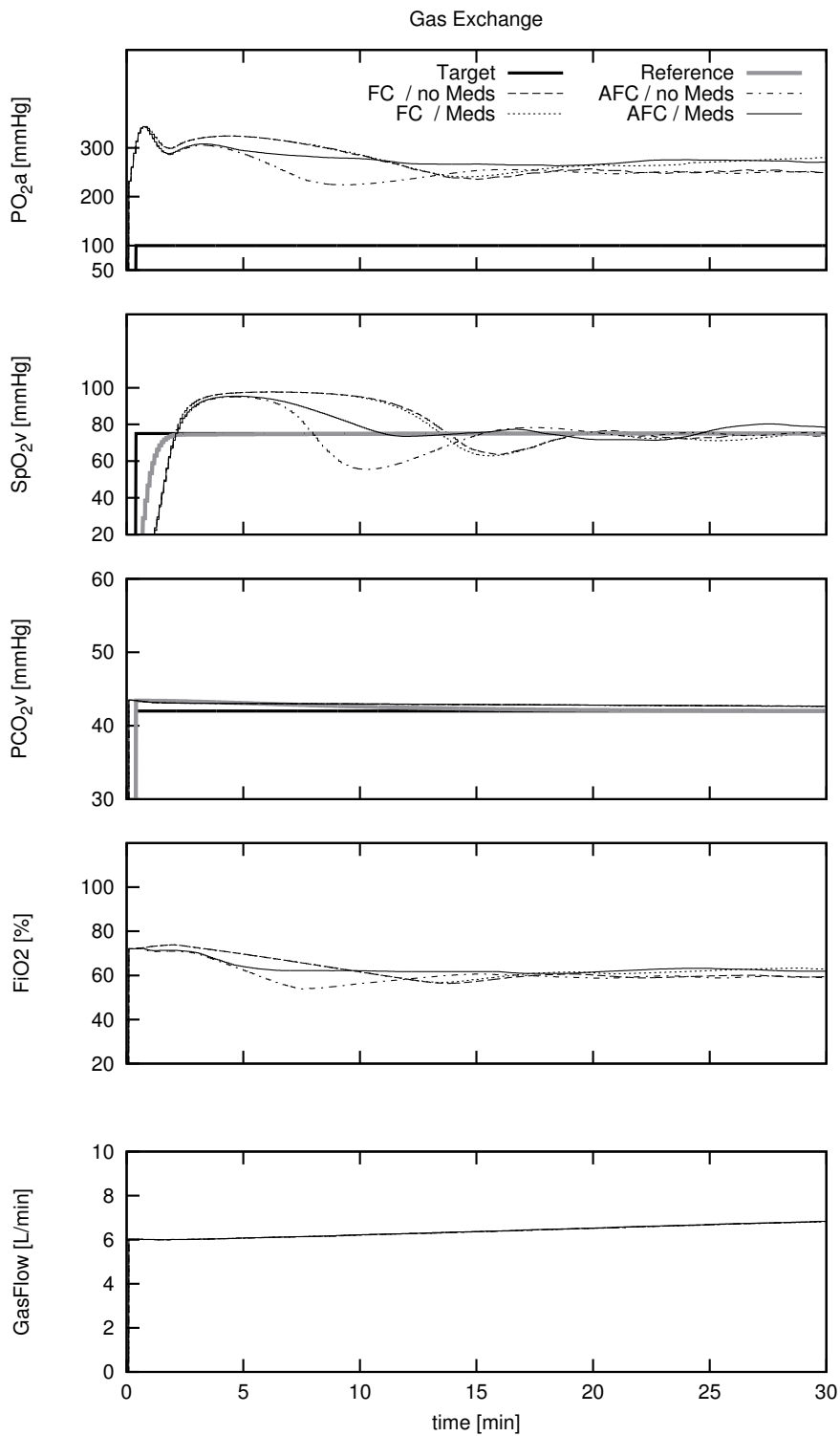


Figure V-7.: Gas results

2.3. Control Comparison

Figure V-8 and V-9 show the accumulative MAE considering all the signals. This was done using the accumulative mean absolute error of the normalized reference errors of each parameter as shown in equation IV.9.

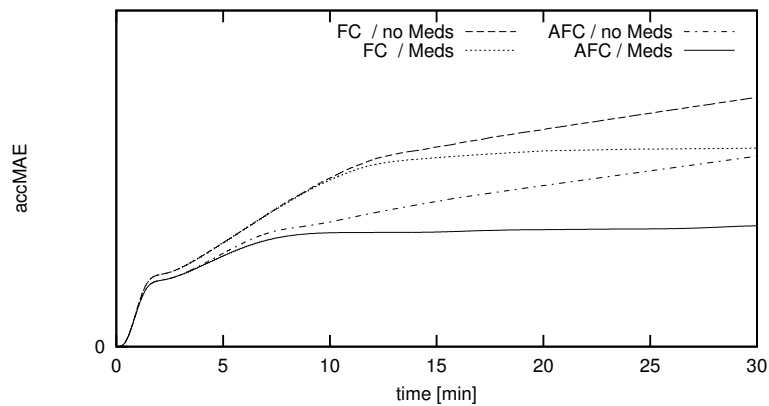


Figure V-8.: Comparison between control mechanisms for first case of high pressure and beating heart.

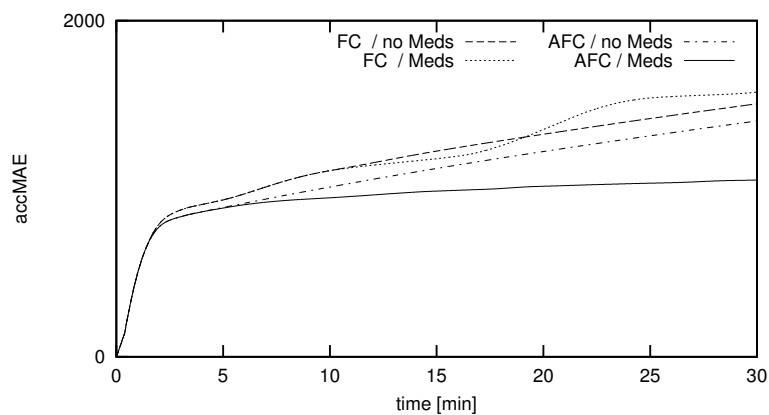


Figure V-9.: Comparison between control mechanisms for second case of low pressure and heart stopped.

These results show that the AFC controllers with medication had better performance than the rest of the controllers. In order to achieve this previous training and simulations were needed. The number of rules created in each controller after adaptation are shown

in table V.1. The rule creation will greatly depend on the number of inputs and number of sets configured for each input. The increased number of rules in the FiO2 and Gas Flow did not affect the controller execution since these controllers are evaluated only every 6 seconds.

Control	Number of rules
Pump Speed	253
FiO2	595
Gas Flow	394
Medication	81

Table V.1.: Number of rules created for each controller after learning and adaptation.

For the cases of control without medication the MAE is constantly increased since the target values can not be reached. When medication was applied oscillations were produced in the case of simple fuzzy controllers exceeding the error caused by the simple FC without medication. The AFC controllers were capable of reducing this oscillations however it was noticed that when the system was close to its target values if the adaptive mechanism is still active small errors may produce rule changes and produce small oscillations; this indicates that the adaptive mechanism should be active only when the inputs are not close to its target.

Chapter VI

Discussion and Conclusions

Contents

1. Discussion	121
2. Conclusion	125
3. Future Work	126

1. Discussion

Extracorporeal circulation has been successfully reported to be used in emergency situations other than cardiac arrest. An example of this was reported by Christoph Eich [35] in a case of saving a drowned child by using extracorporeal membrane oxygenation. Another use of extracorporeal support system is in the aid of victims suffering of hypothermic cardiac arrest. Gregory et al. in 1991 showed that cardiopulmonary bypass is an efficient and physiological method of rewarming hypothermic patients [36]. With a blood temperature gradient of 6 °C and a flow of 4 L/min, the core temperature can rise by up to 10 °C per hour, depending on the size of the patient. [37, p.294] It is likely that patients with severe hypothermia may suffer a cardiac arrest, however using the portable ECSS with an automation system the patients core temperature may be restored while giving adequate perfusion to the body.

Considering the previous examples and the possibility of saving patients suffering from cardiac arrest with the use of an ECC clearly show the importance and advantage of using these systems. Current portable ECSSs need constant adjustments which are based on the continuous patient monitoring and the knowledge and expertise of the operator. Under emergency circumstances however an experienced perfusionist may not always be present and the high level of stress may increase the risk of making wrong decisions. It is evident that an automation system is needed to extend the usability of these devices, making them easier to use, while still being able to guarantee safety and best treatment to the patient.

It is important to mention that extra-corporeal circulation is a procedure of high risk, however it gives patients an opportunity of survival when other procedures have failed. With this system not only should the survival of the patient be considered, but also to

have the best outcome after proper medical treatment has taken place. In this case special emphasis has been given in analyzing what are the best perfusion practices. The studies previously discussed in chapter II section 3 discuss the fact that there is no clinical proof of knowing what is the optimal perfusion that fits all patients; this greatly depends on the patient's previous medical history and current condition. The proposed control mechanism expects the operator to define the target parameters based on his knowledge about the patient and specific hospital practice. Further research could be done to create an autonomous system that considers the current and previous data of the patient and suggest the target parameters that should be achieved.

The improvement of the ECSS is still under constant research. New materials are applied to allow a more extensive use of these devices making them more portable and extending their useful lifetime; new blood pumps and oxygenators may reduce the damage to blood cells and avoid clotting. With the presented characterization of the components in the ECSS it was possible to determine which of them represents a major resistance to the system. This is important since the total resistance of all the components will influence the amount of blood flow that can be generated by the machine. A high component and systemic resistance will generate high pressures when trying to achieve the required flow. This in return can damage the blood cells. The presented results in section III.3.3 showed which components influence the system with a higher resistance providing information on how the system may be improved. The Lifebridge device used during our experiments showed that even with maximum pump speeds the desired pressures and flows could not always be achieved, the subsequent component analysis showed great resistance caused by the arterial and venous cannulas. Current research was found focused on solving this problem by creating self-expanding cannulas [34]. From these new components curves of pressure against flow may be obtained and the results can be introduced in the mathematical model to estimate how the overall system is affected.

Before stepping into the design of the controller it was important to know what signals are acquired during ECC and what sensors are used. The experimental model allowed the evaluation of these signals and gave an understanding of the way the ECSS interacts with the cardiovascular system. From these experiments two distinctive cases were analyzed, one as an interaction between a beating heart and the continuous flow generated by the machine in opposite direction, and the second with the heart completely stopped and the machine fully in charge of organ perfusion. For the first case with of the heart still beating it was shown that the ECSS may reduce the load of the heart, perfusing the trunk and lower extremities, while the heart takes care of the head and upper extremities.

The experiments carried out served as a reference to construct a mathematical model. Describing the model using the MML employed by JSim allowed a clean and easy description of the different subsystems. This made the development of the model faster. One disadvantage that was noticed in the model description was the absence of an object oriented modeling. This could facilitate the model creation by describing a single component in terms of pressure, flow and resistance and easily extending it to a more complex model.

The constructed model together with the developed simulation environment was crucial for the creation of the automation. It allowed the constant evaluation of the control

software and the elimination of possible errors. Additionally it was possible to evaluate different control mechanisms by simulating the same scenario as many times as needed. This type of evaluation and repetition would have not been possible through experimental procedures.

The presented simulation results show the capability of the mathematical model of representing the cardiovascular system of patients with different disorders. The experimental data was used to adjust the model initial parameters; Once a stable system was obtained several conditions such as hypertension or hypotension together with heart illnesses such as mitral valve regurgitation were recreated. These scenarios were based on theoretical data, however further studies may be carried out where real patient data is used to adjust patient parameters and further evaluate the developed controllers.

Fuzzy control is helpful when a mathematical model is difficult to obtain or of high complexity. It provides a systemic and efficient framework for incorporating linguistic fuzzy information from human experts.

In the case of the ECSS, the machine components are fixed, allowing some predictions of the expected flow and pressures, however patient conditions and emergency circumstances will differ from one to another, making it difficult to assume the same mathematical model to be applicable to all the possible scenarios.

If we ask the human operators, in this case the perfusionist, what control strategies they follow they may tell a few control rules in fuzzy terms and some linguistic descriptions about the behavior of the system under various conditions, which are also considered fuzzy. This knowledge is based on their expertise and medical background rather than the full physical and mathematical description of the aspects involving ECC. These fuzzy control rules and descriptions are not precise and may not be sufficient for constructing a fully efficient controller; however they provide important information about how to control the system. With the introduction of several fuzzy controllers it was possible to create a knowledge base system containing this information. These individual controllers contain a general description of how the system should behave focusing in specific aspects such as pressure, flow and oxygen saturation, avoiding the creation of considerably large rule bases. Several rule base structures were presented such as FuzzyPI also simplifying the process of rule creation. One advantage of creating a knowledge base with smaller controllers is that the operator may still choose how much weight is given into the decisions of each controller based on their importance; or they can be completely disabled to focus on a specific kind of control. For example the pump controller can fully work on adjusting the desired MAP without considering EFR or gas consumption. With this mechanism different types of perfusion can be defined.

The perfusionist makes fine adjustments to the ECSS parameters based on the observation of the reactions of the patient to the given treatment, expecting a certain outcome. The adaptive mechanism was proposed to act in the same manner. Additionally the adaptive control was incorporated to maintain a consistent performance of the complete system in the presence of uncertainties. This is done by evaluating the performance of the controller with the use of reference models. These models serve as a predictor on how a system is expected to respond. The information provided by the models and the current state of the system is used to modify the controller rules and adjust the system to each individual

patient.

Previous studies have described a multi-input multi-output adaptive fuzzy/neural control [93]. A disadvantage noticed by this type of control is that by introducing neural networks the system complexity increases, compared to such controller the rules from the adaptive fuzzy controller may still be analyzed and know why specific changes were taken.

Another example of adaptive fuzzy control was shown by Nauck et al. [83, 111] They presented a neuro-fuzzy controller called NEFCON, it is based on a generic fuzzy perceptron. They present two algorithms, one starting with a full rule base and eliminating the rules that are not needed. The drawback of this approach is that when several inputs are included with a considerable number of sets the rule base expands exponentially making it computationally intensive. The second type of algorithm is called 'Bottom-Up' where the controller starts with an empty rule base. The rules are created in a similar way as described in the learning process, however with the absence of the knowledge controllers the algorithm tries to guess the output. The authors mention that this type of algorithm should not be used for complex dynamic systems, due to the heuristic approach of finding the consequents. The presented controller eliminates this problem with the introduction of the knowledge controllers; these smaller controllers contain all the possible combinations of inputs with respect to the defined sets, however by focusing on specific inputs rule bases are kept at a reasonable size.

Additional studies have been carried out in the development of adaptive mechanisms without the use of fuzzy logic. An example of such mechanisms is a robust type of controller using a proportional integral and an H_∞ control proposed by Misgeld et al[88]. The disadvantage of this approach is the use of extensive control formulas that may be difficult to understand for medical doctors reducing the chance of acceptance. Using the adaptive fuzzy controllers it is possible to analyze which rules are applied at every instant and do further corrections when needed.

Concerning the presented implementation, several tools already exists that have the capabilities of fast configuration and easy implementation. Examples of these applications are Simulink from Matlab[127] and Labview[128]. The disadvantage of using these applications is the restrictions of having to use specific compatible hardware and the extension of new components. Embedded code generation is possible; however the verification of such code is difficult making it not recommendable for medical devices.

With the developed software framework it was possible to create an automation system for the ECSS. By using a component based structure and with the definition of a common communication interface individual base objects are easily developed and tested before they are used in a system. This allows parallel development of the components of the system and reduces the risk of failures in case a component needs to be changed. Additionally using configuration files and a graphical representation of the used objects to describe the controller allows an easier understanding of the different components involved without having to analyze extensive lines of programming code.

Besides the ECSS other medical devices with similar characteristics could be also automated with this software framework. Existing components may be used, while still maintaining reliability since the source code remains intact. This can help by significantly

reducing the time required for future developments.

Once the components are verified a complete system can be created which can further be tested via simulations. This can be either with virtual simulations using mathematical models or with mockup hydraulic systems such as the one presented in [125].

The proposed hardware implementation does not have a high demand on hardware requirements and can be easily ported to other platforms if needed. A major concern in embedded systems is power consumption, new technology shows that a way to reduce power, while maintaining processing capabilities, can be done by increasing the number of processors, without significantly increasing clock speed. Current programs are not able to take full advantage of these new processors since the processes may not be easily dissected into multiple threads. This object oriented approach however may allow the easy visualization and creation of parallel processes fully utilizing the available processing resources.

The presented simulation results show that the developed controller was capable of handling several inputs and was able to achieve the target values. This was shown with two scenarios representing different patient pre-conditions.

2. Conclusion

The work presented in this thesis shows the automation of a portable extracorporeal support system meant to be used under emergency circumstances for patients suffering cardiogenic shock outside the hospital.

An experimental model helped to have an idea of the procedure involving extracorporeal circulation and the different sensors that are used to monitor the patient's vital signals. An acquisition system was specifically designed for this application to allow the simultaneous recording and monitoring of the different devices. The acquired data could then be used as a reference for the construction of a mathematical model. Once the designed simulation was capable of representing the experimental data it was possible to proceed with the development of the controller and thoroughly evaluate its performance.

Fuzzy logic was used as a control mechanism. This proved to be an effective mechanism to easily implement the desired control behavior based on the knowledge of the trained perfusionist. Additional adjustments may be required to optimize performance based on specific patient response; an adaptive mechanism was introduced which demonstrated to improve perfusion compared to controllers with no adaptation. The adaptation process uses a set of knowledge controllers together with reference models. These two elements allowed the evaluation of the controller and helped in the creation and modification of control rules.

A software framework was built to easily create a controller for the ECSS. This framework was based on a component based architecture. The framework was ported to an embedded system with low power consumption allowing an easy integration into the ECSS.

3. Future Work

There is still some path to cover before the designed automation may be used in real patients, however the tested simulations show promising results. Additionally with the provided simulation environment it is possible to generate more scenarios that can help in the evaluation and improvement of the controller and its adaptation.

The presented thesis also serves as ground work for the automation of other medical devices where mathematical models may be easily developed and controllers may be implemented and tested using the proposed software framework. Figure VI-1 shows examples of several applications where the knowledge and developed systems presented in this thesis may be applied.

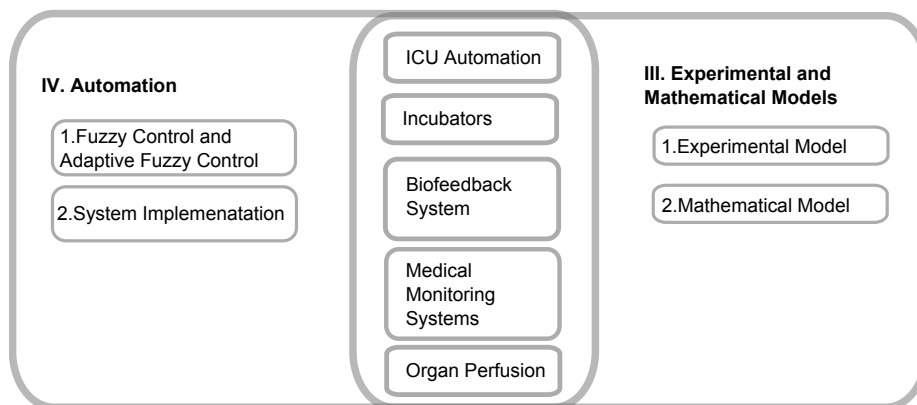


Figure VI-1.: Future applications

ICU Automation

Patients who have gone through a heart operation and are transferred to the intensive care unit may benefit from a monitoring and automation system by monitoring the patient vital signals and providing a gradual change of medication dosage when required. Current research developed at the German Heart Center focuses on mathematical models of the used medications [18].

Incubators

Incubators are other kinds of intensive care units that specialize in the care of ill or premature newborn infants. These devices monitor the newborn patient and are in charge of providing optimal conditions that allow the child's growth. These systems also need control mechanisms that change the incubators parameters such as oxygen and temperature regulation.

Biofeedback System

The goal of this project is to research and develop a system that helps children with heart disorders to recover physical fitness after having an operative treatment. The system uses

electronic games to help motivate the children to perform exercise. Playing games will cause the children to perform moves that are supporting the rehabilitation process. The patient's physical engagement is measured by biomedical sensors and used for controlling the intensity and frequency of moves that are stressed by playing the game. The information of the different sensors is gathered through the developed software framework and a fuzzy controller is used to determine the difficulty of the game.

Medical Monitoring Systems

The portability of the developed framework into different operating systems and its low hardware requirements makes it ideal to develop small systems that are capable of monitoring patients. Fuzzy controllers may be used to diagnose the state of the patient and send data to doctors when needed.

Organ Procurement and Preservation Systems

Once all the possibilities of helping a patient to survive have been exhausted and death is imminent, organ transplant may bring new hope to other patients in need. A small automated perfusion system similar to the ECSS may help to keep specific organs in a good condition while they are taken to the destination hospital for transplantation.

All of the previous examples may involve the use of mathematical models to detect the proper functioning of the system, the presented adaptive fuzzy control mechanisms may be applied in the automation process and the developed software framework brings a fast and reliable way of implementing the different systems.

Bibliography

- [1] Texas Heart Institute. [Online]. Available: <http://texasheart.org/HIC/Anatomy/anatomy2.cfm>
- [2] Pulse oximetry. [Online]. Available: <http://www.oximetry.org/pulseox/principles.htm>
- [3] J. Xu, K. Kochanek, S. L. Murphy, and B. Tejada-Vera, "Deaths: Final data for 2007. national vital statistics reports," *National vital statistics reports*, vol. 58, p. 19, 2010.
- [4] World Health Organization. [Online]. Available: <http://www.who.int/mediacentre/factsheets/fs310/en/index.html>
- [5] U. Satish and S. Streufert, "Value of a cognitive simulation in medicine: towards optimizing decision making performance of healthcare personnel," *Qual Saf Health Care*, vol. 11, pp. 163–167, Jun 2002.
- [6] A. Brunberg, S. Heinke, J. Spillner, R. Autschbach, D. Abel, and S. Leonhardt, "Modeling and simulation of the cardiovascular system: a review of applications, methods, and potentials," *Biomed Tech (Berl)*, vol. 54, pp. 233–244, Oct 2009.
- [7] A. Guyton and J. E. Hall, *Textbook of Medical Physiology*. Saunders, 2005.
- [8] R. G. Carroll, "Cardiovascular pressure-flow relationships: What should be taught?" *ADV PHYSIOL EDUC*, vol. 25, pp. 8–14, 2001.
- [9] Millar instruments. [Online]. Available: <http://www.millarinstruments.com>
- [10] H. Walcher. Calibration equipment:kal84. [Online]. Available: <http://www.halstrup-walcher.com/Details.lasso?Topic1=Presse&Topic2=Kalibriergeraete&ID=140>
- [11] Transonic. [Online]. Available: <http://www.transonic.com>
- [12] Corscience. [Online]. Available: <http://www.corscience.de>
- [13] W. Zong, T. Heldt, G. Moody, and R. Mark, "An open-source algorithm to detect onset of arterial blood pressure pulses," pp. 259 – 262, sept. 2003.

- [14] P. Hamilton, "Open source ecg analysis," pp. 101 – 104, sept. 2002.
- [15] L. Mount and D. Ingram, *The Pig as a Laboratory Animal*. Academic Press, 1971.
- [16] B. Voss, M. Krane, C. Jung, G. Brockmann, S. Braun, T. Gunther, R. Lange, and R. Bauernschmitt, "Cardiopulmonary bypass with physiological flow and pressure curves: pulse is unnecessary!" *Eur J Cardiothorac Surg*, vol. 37, pp. 223–232, Jan 2010.
- [17] A. Mendoza Garcia, B. Baumgartner, U. Schreiber, M. Krane, A. Knoll, and R. Bauernschmitt, "Automedic: Fuzzy control development platform for a mobile heartlung machine." *IFMBE Proceedings*, vol. 25(7), pp. 685–688, 2009.
- [18] N. Sprunk, A. Mendoza Garcia, U. Schreiber, R. Bauernschmitt, and A. Knoll, "Cardiovascular model for development and test of automated hemodynamic regulation with medication," *CINC*, pp. 153 – 156, 2011.

Bibliography Extracorporeal Circulation

- [19] M. Krane, D. Mazzitelli, U. Schreiber, A. M. Garcia, B. Voss, C. Badiu, R. Lange, and R. Bauernschmitt, "First experience with a new portable cardiopulmonary bypass system - lifebridge b2t with percutaneous femoral cannulation," *Comp. In Cardiology*, pp. 269–272, 2008.
- [20] J. H. Stuckey, M. M. Newman, C. Dennis, E. H. Berg, S. E. Goodman, C. C. Fries, K. E. Karlson, M. Blumenfeld, S. W. Weitzner, L. S. Binder, and A. WINSTON, "The use of the heart-lung machine in selected cases of acute myocardial infarction," *Surg Forum*, vol. 8, pp. 342–344, 1957.
- [21] J. Shin, S. Lee, G. Han, W. Jo, S. Choi, and Y. Hong, "Successful extracorporeal life support in cardiac arrest with recurrent ventricular fibrillation unresponsive to standard cardiopulmonary resuscitation." *Resuscitation*, vol. 73(2), pp. 309–13, 2007.
- [22] B. Leshchinskii, G. Itkin, and N. Zimin, "Centrifugal pumps for forcing blood: technical aspects," *Med Tekh.*, vol. 1, pp. 28–31, 1990.
- [23] M. Krane, D. Mazzitelli, U. Schreiber, A. Mendoza, S. Braun, B. Voss, C. Badiu, G. Brockmann, R. Lange, and R. Bauernschmitt., "Lifebridge b2t a new portable cardiopulmonary bypass system." *ASAIO J*, vol. 56, pp. 52–56, 2010.
- [24] G. S. Murphy, E. A. Hessel, and R. C. Groom, "Optimal perfusion during cardiopulmonary bypass: An evidence-based approach," *Anesthesia Analg.*, vol. 108(5), pp. 1394–1417, 2009.
- [25] C. Bartels, A. Gerdes, J. Babin-Ebell, F. Beyersdorf, U. Boeken, T. Doenst, P. Feindt, M. Heiermann, C. Schlensak, and H.-H. Sievers, "Cardiopulmonary bypass: Evidence or experience based?" *Thorac Cardiovasc*, vol. 124, pp. 20–27, 2002.
- [26] M. Ranucci, B. D. Toffol, G. Isgro, F. Romitti, D. Conti, and M. Vicentini, "Hyperlactatemia during cardiopulmonary bypass: determinants and impact on postoperative outcome," *Crit Care.*, vol. 10(6), p. 167, 2006.

- [27] L. E. Cohn, *Cardiac Surgery in the Adult, Third Edition*, E. Lifesciences, Ed. McGraw-Hill, 2008.
- [28] P. Fernandes, J. MacDonald, A. Cleland, G. Walsh, and R. Mayer, "What is optimal flow using a mini-bypass system?" *Perfusion*, vol. 25, p. 133, 2010.
- [29] P. Feindt, C. Benk, U. Boeken, A. Bauer, U. Mehlhorn, J. Gehron, A. Markewitz, A. Beckmann, and F. Beyersdorf, "Use of extracorporeal circulation (ecc) outside the cardiac operating room: Indications, requirements and recommendations for routine practice," *Thorac Cardiovasc Surg*, vol. 59(2), pp. 66–68, 2011.
- [30] A. V. Govier, J. G. Reves, R. D. McKay, R. B. Karp, G. L. Zorn, R. B. Morawetz, L. R. Smith, M. Adams, and A. M. Freeman, "Factors and their influence on regional cerebral blood flow during nonpulsatile cardiopulmonary bypass," *Ann. Thorac. Surg.*, vol. 38, pp. 592–600, Dec 1984.
- [31] C. O Dwyer, L. C. Woodson, B. P. Conroy, C. Y. Lin, D. J. Deyo, T. Uchida, and W. E. Johnston, "Regional perfusion abnormalities with phenylephrine during normothermic bypass," *Ann. Thorac. Surg.*, vol. 63, pp. 728–735, Mar 1997.
- [32] R. J. Gordon, M. Ravin, R. E. Rawitscher, and G. R. Daicoff, "Changes in arterial pressure, viscosity and resistance during cardiopulmonary bypass," *J. Thorac. Cardiovasc. Surg.*, vol. 69, pp. 552–561, Apr 1975.
- [33] F. Contreras, C. Fouillioux, A. Bolivar, N. Simonovis, R. Hernandez, M. A. Hernandez, and M. Velasco, "Dopamine, hypertension and obesity," *Human Hypertension*, vol. 16, pp. 13–17, 2002.
- [34] L. K. von Segesser, M. Kalejs, E. Ferrari, S. Bommeli, O. Maunz, J. Horisberger, and P. Tozzi, "Superior flow for bridge to life with self-expanding venous cannulas," *Eur J Cardiothorac Surg*, vol. 36, pp. 665–669, Oct 2009.
- [35] C. Eich, A. Brauer, and D. Kettler, "Recovery of a hypothermic drowned child after resuscitation with cardiopulmonary bypass followed by prolonged extracorporeal membrane oxygenation." *Resuscitation*, vol. 67, p. 145, 2005.
- [36] J. S. Gregory, L. Flanckbaum, W. L. Smead, T. E. Reilley, and O. Jonasson, "Extracorporeal venovenous recirculation for the treatment of hypothermia during elective aortic surgery: a phase I study," *Surgery*, vol. 114, pp. 40–45, Jul 1993.
- [37] P. H. Kay and C. Munsch, *Techniques in Extracorporeal Circulation*. Arnold, London, 2004.

Bibliography Mathematical Model

- [38] J. McLeod, "Physbe a physiological simulator." *Simulation*, vol. 7(6), pp. 324–329, 1966.
- [39] J. Bassingthwaite, P. Hunter, and D. Noble, "The cardiac physiome: perspectives for the future," *Exp Physiol*, vol. 5, pp. 597–606, 1994.
- [40] K. Uemura, A. Kamiya, S. Shimizu, T. Shishido, M. Sugimachi, , and K. Sunagawa, "Comprehensive physiological cardiovascular model enables automatic correction of hemodynamics in patients with acute life-threatening heart failure," in *28th IEEE EMBS Annual International Conference*, 2006.
- [41] L. Sheffer, W. Santamore, and O. Barnea, "Cardiovascular simulation toolbox," *Cardiovas*, vol. 7, pp. 81–88, 2007.
- [42] M. Tawhai, J. Bischoff, D. Einstein, A. Erdemir, T. Guess, and J. Reinbolt, "Multiscale modeling in computational biomechanics." *IEEE Engineering in Medicine and Biology Magazine*, vol. 28, no. 3, pp. 41–49, 2009. [Online]. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2854014&tool=pmcentrez&rendertype=abstract>
- [43] R. Bauernschmitt, E. Naujokat, H. Mehmanesh, S. Schulz, C. Vahl, S. Hagl, and R. Lange, "Mathematical modeling of extracorporeal circulation: Simulation of different perfusion regimens perfusion," *Perfusion*, vol. 14, pp. 321–330, 1999.
- [44] J. Dagan, "Pulsatile mechanical and mathematical model of the cardiovascular system," *Med & Biol Eng & Comput*, vol. 20, pp. 601–607, 1982.
- [45] M. Hassan, M. El-Brawany, and M.M.Sharaf, "A functional cardiovascular model with disorders," *Engineering in Medicine and Biology*, vol. 27, pp. 5089–5092, 2005.
- [46] C.Luo, D. Ware, J. Zwischenberger, and J. C. Jr., "Using a human cardiopulmonary model to study and predict normal and diseased ventricular mechanics, septal interaction, and atrio-ventricular blood flow patterns," *Cardiovascular Engineering*, vol. 7:17, p. 31, 2007.

- [47] M. L. Neal and J. B. Bassingthwaighte, "Subject-specific model estimation of cardiac output and blood volume during hemorrhage," *Cardiovasc Eng*, vol. 7, pp. 97–120, 2007.
- [48] J. Olansen, J. Clark, D. Khoury, F. Ghorbel, and A. Bidani, "A closed-loop model of the canine cardiovascular system that includes ventricular interaction," *Am J Physiol Heart Circ Physiol*, vol. 281, pp. 2661–2679, 2001.
- [49] E. B. Shim, J. Y. Sah, and C. H. Youn, "Mathematical modeling of cardiovascular system dynamics using a lumped parameter method," *japanese journal of physiology*, vol. 54, pp. 545–53, 2004.
- [50] B. W. Smith., J. G. Chase., G. M. Shaw., and R. I. Nokes., "Simulating transient ventricular interaction using a minimal cardiovascular system model," *Physiological Measurement*, vol. 27, pp. 165–179, 2006.
- [51] K. Lu, J. W. Clark, F. Ghorbel, D. Ware, and A. Bidani, "An integrated model of the human cardiopulmonary system," in *EMBC*, 2001.
- [52] A. Mendoza, B. Baumgartner, U. Schreiber, M. Krane, R. Bauernschmitt, and A. Knoll, "Simulation of extracorporeal circulation for the design of a fuzzy controlled perfusion." *IASTED Biomedical Engineering*, vol. 1,2, 2010.
- [53] R. Bauernschmitt, S. Schulz, H. Mahmanesh, C. Vahl, and R. Lange, "Simulation of baroreflex control in a pulsatile mathematical model of the human arterial circulation," *Computers in Cardiology*, vol. 26, pp. 229–232, 1999.
- [54] K. Lu, J. W. J. Clark, F. H. Ghorbel, D. L. Ware, J. B. Zwischenberger, and A. Bidani, "Whole-body gas exchange in human predicted by a cardiopulmonary model," *Cardiovascular Engineering*, vol. 3, pp. 1–19, 2003, 10.1023/A:1024795417999. [Online]. Available: <http://dx.doi.org/10.1023/A:1024795417999>
- [55] T. Davis, "Teaching physiology through interactive. simulation of hemodynamics," *Computers in Cardiology*, 1991.
- [56] H. Chizeck, E. Butterworth, and J. B. Bassingthwaighte, "Error detection and unit conversion: Automated unit balancing in modeling interface systems," *IEEE Engr. Med. Biology Mag*, vol. 28(3), pp. 50–58, 2009.
- [57] N. Watanabe, O. Karsak, F. Neudel, T. Kink, J. Apel, T. Fujimoto, H. Reul, and S. Takatani, "Simulation of the bp-80 blood pump," *Artificial Organ*, vol. 25, pp. 733–39, 2001.
- [58] K. Lu, J. C. Jr., F. Ghorbel, D. Ware, and A. Bidani, "A human cardiopulmonary system model applied to the analysis of the valsava maneuver," *Physiol Heart Circ*, vol. 281, pp. 2661–2679, 2001.
- [59] S. JW, K. P, and G. E., "Transfer characteristics of the carotid sinus pressure control system." *Baroreceptors and Hypertension*, pp. 31–40, 1965.

- [60] M. Ursino, "Interaction between carotid baroregulation and the pulsating heart: A mathematical model." *Am. J. Physiol., Heart Circu. Physiol.*, vol. 275(5), pp. 1733–1747, 1998.
- [61] Brawany, "A functional cardiovascular model with iv cardiac drugs action," *Electronics and Electrical Engineering*, vol. 1(1), pp. 35–42, 2009.
- [62] K. Brandis, *The Physiology Viva*.
- [63] R. H. Clayton, D. T. Pearson, and A. Murray, "Assessment of oxygen transfer in membrane oxygenators during clinical cardiopulmonary bypass," *Clin. Phys. Physiol. Meas.*, vol. 13(2), pp. 167–177, 1992.
- [64] W. J. Federspiel, T. J. Hewitt, and B. G. Hattler, "Experimental evaluation of a model for oxygen exchange in a pulsating intravascular artificial lung," *Annals of Biomedical Engineering*, vol. 28, pp. 160–167, 2000.
- [65] E. P. Hill, G. G. Power, and L. D. Longo, "Mathematical simulaton of pulmonary o2 and co2 exchange," *American Journal of Physiology*, vol. 224, p. 4, 1973.
- [66] P. Huang, S. Nioka, and B. Chance, "A four compartment model of oxygen delivery to muscle tissue from capillary," *Bioengineering Conference*, pp. 259–260, 2002.
- [67] B. J. Misgeld, J. Werner, and M. Hexamer, "Nonlinear robust blood-gas control by state linearisation for the cardiopulmonary bypass," *Control Engineering Practice*, vol. 16, pp. 884–896, 2008.
- [68] M. Sharan, M. Singh, and A. Aminatei, "A mathematical model for the computation of oxygen disocciation curve in human blood," *Bio.Systems*, pp. 249–260, 1989.
- [69] L. Smeby and L. Grimsrud, "Theoretical investigation of mass transfer in membrane oxygenators," *Medical and Biological Engineering*, pp. 698–706, 1974.
- [70] J. Zhang, T. D. Nolan, T. Zhang, B. Griffith, and Z. J. Wu, "Characterization of membrane blood oxygenation devices using computational fluid dynamics," *Membrane Science*, vol. 288, pp. 268–279, 2007.
- [71] P. Rand, E. Lacombe, and H. Hunt, "Viscosity of normal human blood under normothermic and hypothermic conditions," *Appl Physiol.*, vol. 19, pp. 117–122, 1964.
- [72] C. Liu, S. Niranjana, J. C. Jr, K. San, J. Zwischenberger, and A. Bidani, "Airway mechanics, gas exchange, and blood flow in a nonlinear model of the normal human lung." *Appl Physiol.*, vol. 84(4), pp. 1447–69, 1998.
- [73] R. Isermann, *Mechatronic Systems: Fundamentals*. Springer, 2005.
- [74] "Jsim." [Online]. Available: <http://nsr.bioeng.washington.edu/jsim/>

Bibliography Control

- [75] R. Bauernschmitt, J. Hoerer, E. Schirmbeck, H. Keil, G. Schrott, A. Knoll, and R. Lange, "Fuzzy-logic based automatic control of hemodynamics," in *Computers in Cardiology, 2003*, sept. 2003, pp. 773 – 776.
- [76] J. H. T. Bates and M. P. Young, "Applying fuzzy logic to medical decision making in the intensive care unit," *Am. J. Respir. Crit. Care Med.*, vol. 167, no. 7, pp. 948–952, 2003. [Online]. Available: <http://ajrccm.atsjournals.org>
- [77] C. W. Chan, K. C. Cheung, and W. K. Yeung, "A computation-efficient on-line training algorithm for neurofuzzy networks." *Int. J. System Science*, pp. 297–306, 2000.
- [78] B. Kosko, *Fuzzy Thinking: The New Science of Fuzzy Logic*. Hyperion Books, 1993.
- [79] H.-X. Li and H. Gatland, "Conventional fuzzy control and its enhancement," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 26, no. 5, pp. 791–797, oct 1996.
- [80] D. G. Mason, D. A. Linkens, and N. D. Edwards, "Self-learning fuzzy logic control in medicine," in *Proceedings of the 6th Conference on Artificial Intelligence in Medicine in Europe*, ser. AIME '97. London, UK: Springer-Verlag, 1997, pp. 300–303. [Online]. Available: <http://dl.acm.org/citation.cfm?id=648154.751451>
- [81] F. Mendi, K. Boran, and M. K. Kulekci, "Fuzzy controlled central heating system," *International Journal of Energy Research*, vol. 26, no. 15, pp. 1313–1322, 2002.
- [82] T. Nazmy, H. El-Messiry, and B. Al-Bokhity, "Adaptive neuro-fuzzy inference system for classification of ecg signals," in *Informatics and Systems (INFOS), 2010 The 7th International Conference on*, march 2010, pp. 1 –6.
- [83] A. Nurnberger, D. Nauck, and R. Kruse, "Neuro-fuzzy control based on the nefcon-model under matlab/simulink," *SOFT COMPUTING*, vol. 2, pp. 168–182, 1997.
- [84] M. Suzer and K.-D. Kang, "Adaptive fuzzy control for utilization management," in *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, may 2008, pp. 383 –390.

-
- [85] W.-M. Lippe and S. Niendieck, "Different methods for the fine-optimization of fuzzy-rule-based-systems," in *Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on*, vol. 2, 2002, pp. 1210–1215.
- [86] A. Abraham, "Adaptation of fuzzy inference system using neural learning," 2005.
- [87] C. Koutsojannis and I. Hatzilygeroudis, "Using a neurofuzzy approach in a medical application." in *KES (2)'07*, 2007, pp. 477–484.
- [88] B. Misgeld, J. Werner, and M. Hexamer, "Robust and self-tuning blood flow control during extracorporeal circulation in the presence of system parameter uncertainties," *Medical and Biological Engineering and Computing*, vol. 43, pp. 589–598, 2005.
- [89] L.-X. Wang, "Stable adaptive fuzzy control of nonlinear systems," in *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, 1992, pp. 2511–2516 vol.3.
- [90] H.-X. Li, "Approximate model reference adaptive mechanism for nominal gain design of fuzzy control system," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 29, no. 1, pp. 41–46, feb 1999.
- [91] M. Hexamer, J. Werner, and B. J. E. Misgeld, "Concepts for simplifying automatic blood-gas control during extracorporeal circulation," in *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany*, ser. IFMBE Proceedings, O. D'Ásel, W. C. Schlegel, and R. Magjarevic, Eds. Springer Berlin Heidelberg, 2009, vol. 25/7, pp. 550–553.
- [92] M. J. Er and Y. Gao, "Adaptive control strategy for blood pressure regulation using a fuzzy neural network," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 3, oct. 2003, pp. 2120–2125 vol.3.
- [93] R. Ordonez and K. Passino, "Stable multi-input multi-output adaptive fuzzy/neural control," *Fuzzy Systems, IEEE Transactions on*, vol. 7, no. 3, pp. 345–353, jun 1999.
- [94] A. Mendoza G, B. Baumgartner, U. Schreiber, S. Eichhorn, M. Krane, R. Bauernschmitt, and A. Knoll, "Design of a fuzzy controller for the automation of an extra-corporeal support system with the use of a simulation environment," pp. 6698–6701, 31 2010-sept. 4 2010.
- [95] B. Baumgartner, A. Mendoza, U. Schreiber, S. Eichhorn, M. Krane, R. Bauernschmitt, and A. Knoll, "A 4-layer supervising unit for extra-corporal circulation," in *Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference on*, june 2010, pp. 1–4.
- [96] B. Baumgartner, A. Mendoza G, U. Schreiber, S. Eichhorn, M. Krane, R. Bauernschmitt, and A. Knoll, "A comprehensive approach towards extra-corporal circulation control using fuzzy logic," pp. 1–5, july 2010.
- [97] U. Schreiber, S. Eichhorn, A. Mendoza, B. Baumgartner, R. Bauernschmitt, R. Lange, A. Knoll, and M. Krane, "A new fuzzy controlled extracorporeal circulation system. first results of an in-vitro investigation," *Computers in Cardiology*, pp. 497–500, 2009.

- [98] T. Nemoto, G. E. Hatzakis, C. W. Thorpe, R. Olivenstein, S. Dial, and J. H. T. Bates, "Automatic control of pressure support mechanical ventilation using fuzzy logic," *Am. J. Respir. Crit. Care Med.*, vol. 160, pp. 550–556, 1999.
- [99] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.
- [100] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Addison-Wesley, 1998.
- [101] H.-X. Li, "Adaptive fuzzy control," *Control Laboratory Department of Electrical Engineering*, pp. 366–371, 1996.
- [102] S. Merz, R. Bartlett, J. Jenkins, and P. Kabamba, "Controller design for extracorporeal life support," in *Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine. Proceedings of the 18th Annual International Conference of the IEEE*, vol. 4, oct-3 nov 1996, pp. 1733–1735 vol.4.
- [103] J. Allen, A. C. Fisher, J. D. Gaylor, and A. R. Razieh, "Development of a digital adaptive control system for PO₂ regulation in a membrane oxygenator," *J Biomed Eng*, vol. 14, pp. 404–411, Sep 1992.
- [104] R. E. Dugdale, R. G. Cameron, and G. T. Lealman, "Closed-loop control of the partial pressure of arterial oxygen in neonates," *Clin Phys Physiol Meas*, vol. 9, pp. 291–305, Nov 1988.
- [105] P. Theunissen, "Regulation of arterial oxygen partial pressure during cardio pulmonary bypass," Master's thesis, University of Maastricht, Instrumental Department., 1997.
- [106] B. Aufderheide, R. Rao, and B. Bequette, "Automated hemodynamic regulation with model predictive control," in *Bioengineering Conference, 1999. Proceedings of the IEEE 25th Annual Northeast*, apr 1999, pp. 73–74.
- [107] R. Rao, C. Palerm, B. Aufderheide, and B. Bequette, "Automated regulation of hemodynamic variables," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 20, no. 1, pp. 24–38, jan.-feb. 2001.
- [108] R. Rao, B. Aufderheide, and B. Bequette, "Experimental studies on multiple-model predictive control for automated regulation of hemodynamic variables," *Biomedical Engineering, IEEE Transactions on*, vol. 50, no. 3, pp. 277–288, march 2003.
- [109] B. Baumgartner, A. Mendoza Garcia, U. Schreiber, S. Eichhorn, M. Krane, R. Bauernschmitt, and A. Knoll, "A simple fuzzy controller for an extra-corporeal circulation system - limitations and potentials." *5th Russian Bavarian Congress Proceedings*, 2009.
- [110] B. J. Misgeld, J. Werner, and M. Hexamer, "Nonlinear robust blood gas control by state linearisation for the cardiopulmonary bypass," *Control Engineering Practice*, vol. 16, no. 7, pp. 884–895, 2008.
- [111] D. Nauck, "A fuzzy perceptron as a generic model for neuro-fuzzy approaches," in *In Proc. of the 2nd German GI-Workshop Fuzzy-Systeme '94, Munich*, 1994.

Bibliography Implementation

- [112] Control and measurement interface. [Online]. Available: <http://www.comedi.org/>
- [113] Qt. Embedded linux. [Online]. Available: <http://qt.nokia.com>
- [114] A. Mendoza Garcia, B. Baumgartner, U. Schreiber, M. Krane, A. Knoll, and R. Bauernschmitt, "Automedic: Fuzzy control development platform for a mobile heartlung machine." *IFMBE Proceedings*, vol. 25(7), pp. 685–688, 2009.
- [115] National instruments. [Online]. Available: <http://www.ni.com>
- [116] Freescale. MCF51QE128 evaluation board. [Online]. Available: <http://www.freescale.com>
- [117] Keil, "MCB2300 evaluation board." [Online]. Available: <http://www.keil.com/mcb2300/>
- [118] IAR. STR912FA development board. [Online]. Available: <http://www.iar.com>
- [119] "Technologic Systems. TS-7300." [Online]. Available: http://www.ti.com/lscs/ti/microcontroller/16-bit_msp430
- [120] Texas instruments msp430. [Online]. Available: <http://www.ni.com>
- [121] E. A. Lee, "Embedded software," *Advances in Computers*, vol. 56, 2002.
- [122] J. Justiniano and V. Gopaldaswamy, *Practical design control implementation for medical devices*. Interpharm/CRC, 2003.
- [123] A. M. Ibrahim, *Fuzzy Logic for Embedded Systems Applications*. Elsevier, 2003.
- [124] R. Barry, *Using the FreeRTOS Real Time Kernel - A Practical Guide*. Real Time Engineers, 2009.
- [125] U. Schreiber, S. Eichhorn, A. Mendoza, B. Baumgartner, R. Bauernschmitt, R. Lange, A. Knoll, and M. Krane, "A new fuzzy controlled extracorporeal circulation system. first results of an in-vitro investigation," *CINC*, vol. 36, pp. 497–500, 2009.

- [126] A. Mendoza G, M. Rodriguez H, B. Baumgartner, U. Schreiber, and A. Knoll, "Embedded platform for automation of medical devices," *Computers in Cardiology*, 2011.
- [127] MathWorks. Simulink, matlab. [Online]. Available: <http://www.mathworks.de/products/matlab/>
- [128] National Instruments. Labview. [Online]. Available: <http://www.ni.com/labview/>

Appendix **A**

Detailed Descriptions

1. Abbreviations

BPM	Beats per minute.
BSA	Body surface area.
C_n	Compliance of compartment n .
CBF	Cerebral blood flow.
CPB	Cardiopulmonary bypass.
CPR	Cardiopulmonary resuscitation.
CVP	Central Venous Pressure.
E	Elastance.
EFR	Extracorporeal flow rate.
ECSS	Extracorporeal circulatory support system.
ECC	Extracorporeal circulation.
ECG	Electrocardiogram.
EEG	Electroencephalogram.
FAA	Flow Aorta Ascendens.
FAD	Flow Aorta Descendens.
Hcrit	Hematocrit.
HR	Heart rate.
L	Inductance.
MAP	Mean Arterial Pressure.
MDC	Medical Device Control.
mmHg	millimeters Mercury.

NEP	Norepinephrine vasoconstrictor drug.
PCO _{2a}	Arterial carbon dioxide partial pressure.
PCO _{2v}	Venous carbon dioxide partial pressure.
PO _{2a}	Arterial oxygen partial pressure.
PO _{2v}	Venous oxygen partial pressure.
R _n	Resistance of compartment <i>n</i>
SNP	Sodiumnitroprussid vasodilator drug.
SpO _{2a}	Arterial oxygen saturation.
SpO _{2v}	Venous oxygen saturation.

2. Programs

The following is a list of the programs and tools that were used throughout the development of the work presented in this thesis.

Comedi	Linux control and measurement device interface http://www.comedi.org/
DIA	Program used for creation of UML diagrams. http://projects.gnome.org/dia/
Eclipse	Multi-language software development environment comprising an integrated development environment (IDE) http://www.eclipse.org/
Eureqa	Software tool for detecting equations and mathematical relationships in data sets. http://creativemachines.cornell.edu/eureqa
FreeRTOS	Free real-time operating system http://www.freertos.org/
Gnuplot	Multiplatform portable command-line driven graphing utility http://www.gnuplot.info/

Inkscape	Open Source vector graphics editor http://inkscape.org/
JSim	Java-based simulation system for building quantitative numeric models. http://www.physiome.org/jsim/
NIDaq	National instruments Data acquisition driver http://www.ni.com/dataacquisition/nidaqmx.htm
Qt	http://qt.nokia.com/
QtCreator	http://qt.nokia.com/
WxWidgets	http://www.wxwidgets.org/

Appendix B

Developed Software

This part of the appendix describes in more detail the software implemented for the work of this thesis. The first section explains the software libraries developed for data acquisition and control. These libraries provide different components that are integrated into the software framework described in chapter IV section 3. The second section focuses on the description of the simulation system that provides a continuous simulation of the model described in chapter III section 2.

1. AutoMedic Libraries

Figure B-1 shows an overview of the AutoMedic libraries.

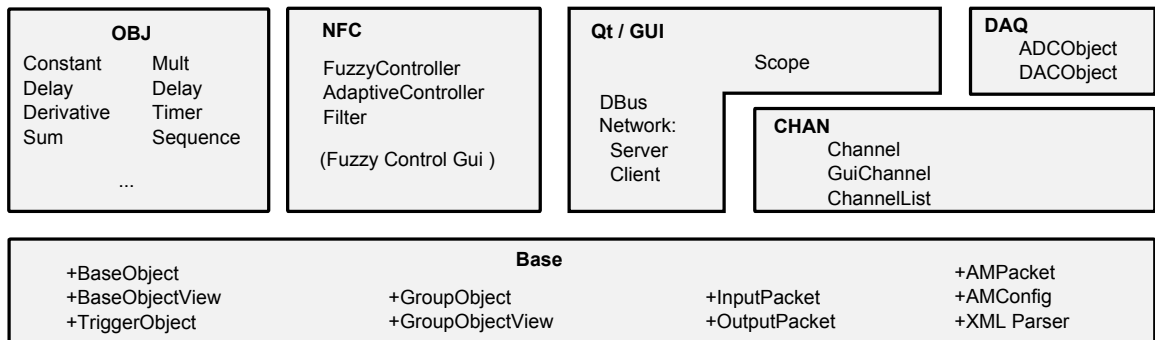


Figure B-1.: Object Libraries

1.1. Base

The Base library contains the main classes that are shared by the rest of the libraries. This includes the BaseObject class from which all of the components in the software framework derive from. The BaseObject was previously described in IV.3.4 together with the

BaseObjectView and TriggerObject. Additional to these classes is the XML parser to read the component parameters from the configuration files and the AMPacket class defined for network intercommunication.

AMPacket

The AMPacket is used as a generic packet of different types of data that can be serialized and transmitted through network or serial interfaces. It is also used for the storage of raw data. This packets may be quickly parsed and used by the upper layers. The basic structure is described bellow:

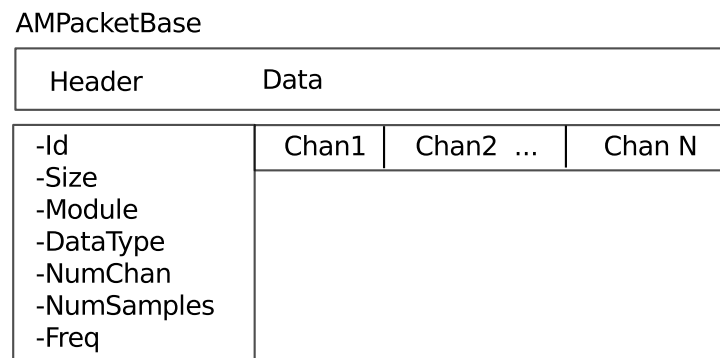


Figure B-2.: AMPacket Structure.

The packet contains a header indicating the Id, defined as 0x24 to identify the packet, the size, a module identifier to know from which device the packet was generated, a DataType defined as 32 bit signed integer or float, the number of channels in the packet and the number of samples contained in each packet. Additionally a frequency specifies what is the time interval between each sample.

AMConfig

This class used to store configuration settings, based on XML, this consists of a simple parser, Qt XML parser was not used to leave this library without dependencies.

1.2. CHAN

This library provides channels with dynamic capabilities. Once a channel is initialized it may be extended indefinitely without caring of memory allocation. A channel is initially configured to a specific frequency. Data may be accessed with a specific position or a given time, internally the channel class solves the corresponding position according to frequency. Data is stored in frames, when a frame is full, memory is allocated for a new frame. The channel may be configured to set the frame size and the maximum number of frames that may be allocated, once the maximum number of frames is reached the last frame is erased, taking care that it is not being accessed. Figure B-3 shows an example of this frame mechanism with frame removal. A GuiChannel class is provided to add additional

data to the AMChannel for visualization purposes. This class contains information such as color, scale and offset. The GuiChannel contains a pointer to an AMChannel giving the possibility to have several GuiChannels pointing to the same AMChannel data with different graphical representations.

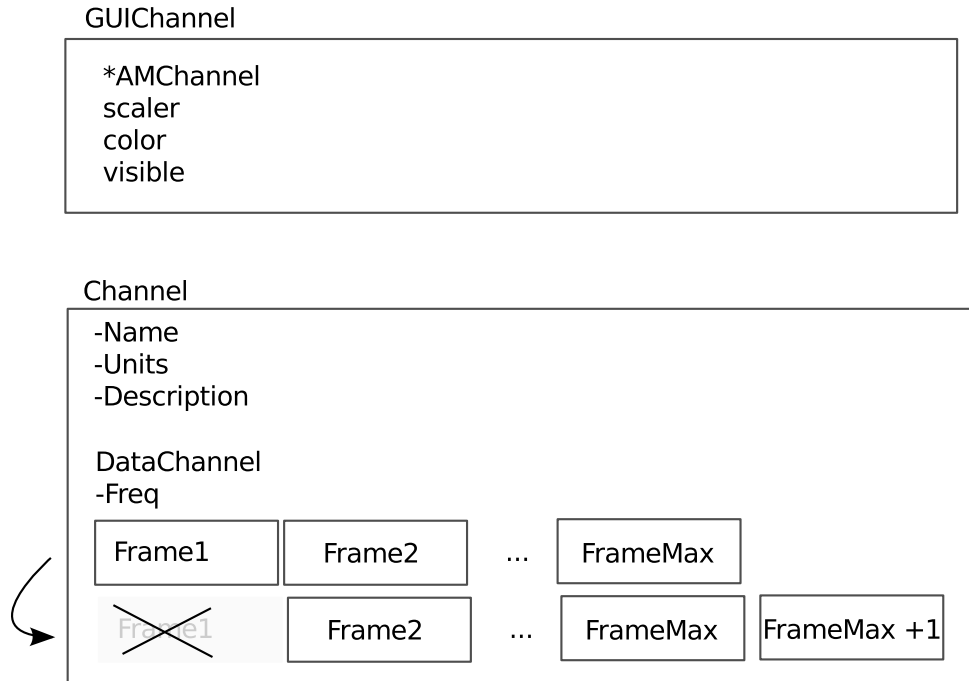


Figure B-3.: Dynamic Channels.

A ChannelList and GuiChannelList provides a way to group all channels and modify group settings.

1.3. Qt / GUI

The Qt/GUI is composed of components that depend on the Qt libraries. This can be for the creation of user interfaces or platform independent components such as network connectivity. This library contains a Scope object which points to a GuiChannel containing data, and visualization information.

The previous figure shows a basic scope with three signals. The user may select a specific time to read values, hide channels and select specific intervals to export to csv files.

1.4. OBJ

This library contains different different general purpose objects. Table B.1 refers to the implemented objects in this module.

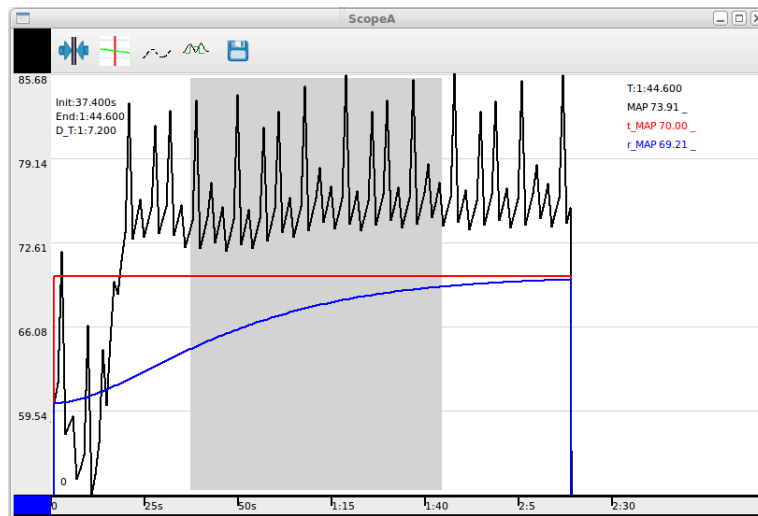


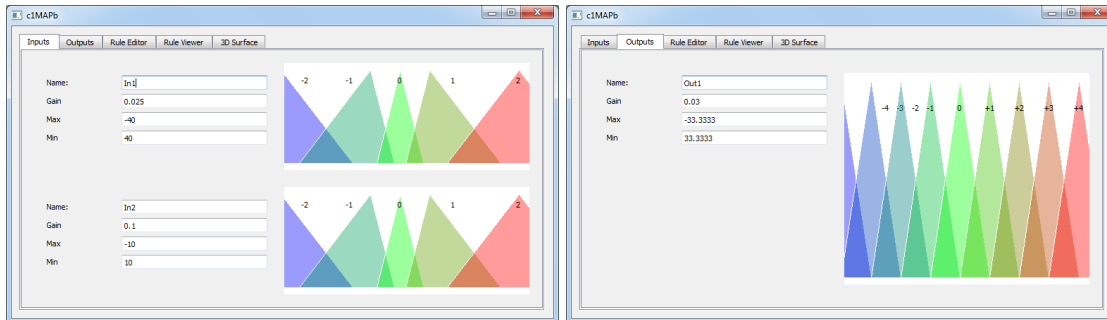
Figure B-4.: Scope

Object Name	Description
Constant	Outputs a constant value.
Delay	Stores N values and after a specific delay outputs value t-delay.
Derivative	Calculates the derivative of the input value.
Gain	Multiplies the input value by a constant gain.
If	Outputs depend on condition.
InputSwitch	Output depends on active input.
Inverse	Calculates the inverse value of the input given by $Out=1/In$
MSE	Calculates the Mean Square Error of a given number of inputs.
MAE	Calculates the Mean Absolute Error of a given number of inputs.
Mul	Multiplies the inputs.
InputPacket	Receives input packets from the network.
OutputPacket	Sends packets to the network containing the values of the given inputs.
Saturation	Sets maximum and minimum values to the input.
Sequence	Specifies a sequence of outputs with a specified duration.
Sum	Calculates the sum of the given inputs, including its weights.
Timer	Generates a trigger signal to the application.
ValueSlider	Creates a slider where the user may introduce the value of the output.
Filter	Filters the input signal with a given frequency.
InParameter	Defines a parameter to use including maximum, minimum.
ReferenceModel	Generates a reference signal with a second order value.
Logger	Saves in a comma separated text file the inputs of this object.

Table B.1.: Implemented Objects, derived from BaseObject

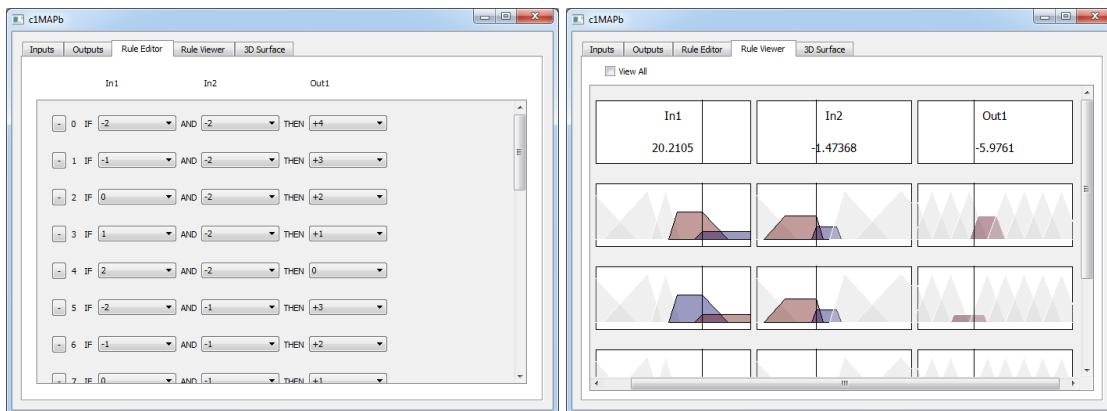
1.5. NFC

This library contains a basic *FuzzyControl* object and an *AdaptiveFuzzyControl* object. The fuzzy controller may be defined with a specific number of inputs and outputs. A user interface may be used to modify the settings of the controller as shown in figure B.



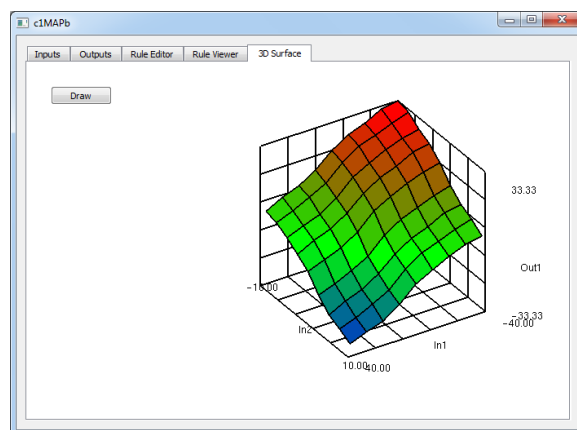
(a) Inputs Panel

(b) Outputs Panel



(c) Rule Editor

(d) Rule Viewer



(e) 3D Surface Viewer

Figure B-5.: Fuzzy Control Viewer

In the Inputs and Outputs panel the sets for each variable may be modified. The Surface panel shows how the output control surface looks for two inputs. In the Rule Viewer it is possible to visualize the active rules of the controller and how they influence the output. In the Rule Editor rules may be added, deleted and modified dynamically. Changes of the controller are saved in text files. The Adaptive Fuzzy Controller works by first integrating a learning part of how the controller should work and an adaptive part where the controller modifies its rule base to improve performance based on a Fuzzy Inverse Model and calculating the Mean Absolute Error of the input parameters.

1.6. DAQ

This library was defined for devices with data acquisition and digital/analog output capabilities. Generic interfaces were created to define the functions that should be implemented by each device driver. On top of these device interfaces visualization panels were created to be able to visualize, configure and calibrate the input and output signals. The structure of the device drivers was previously shown in chapter IV figure IV-17

ADC_DeviceInt	Implemented by devices capable of analog to digital conversion.
DAC_DeviceInt	Implemented by devices capable of digital to analog conversion.
DIO_DeviceInt	Implemented by devices capable of generating and capturing digital signals.
InterfaceListener	This class is implemented by classes that want to listen when one of the devices from the previous interfaces has new data.

Table B.2.: Component Interfaces

ADC7300	PC100	16 analog input signals, 2 analog output signals. ARM Architecture with Linux and Embedded QT
NIDaq6036	PCMCIA	16 16bit analog input signals, 2 analog output signals Under Linux it uses the Comedi driver Under Windows it uses the NIDaq driver provided by National Instruments

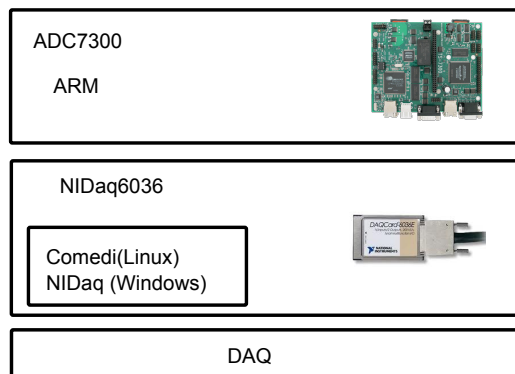


Figure B-6.: Daq device drivers

1.7. SerialCom

The following device drivers were integrated into the system. The devices communicating through a serial interface such as bluetooth, virtual serial port through USB and RS232 depend on the library SerialCom. Each driver library extends a base object to be able to integrate the device with the rest of the components.

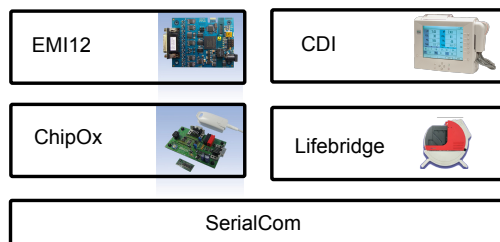


Figure B-7.: Serial device drivers

Device	Interface	Data acquired
ChipOx	RS232	Plethysmogram, Heart rate, Oxygen Saturation
EMI12	USB	ECG (200Hz,500Hz) (3,6,12 Lead) Heart rate with HB lib
CDI	RS232	Gas Analyzer

2. Simulation

This section describes the different components of the simulation environment. The simulation system was implemented on top of the JSim simulation system [74] as shown on figure III-31 on section III.3.

2.1. Simulation Manager

The simulation manager is the main component of the simulation application. Its in charge of loading the mathematical model written in MML and compiling it. This is associated with a SimEngine in charge of starting/stopping and running the simulation. Additional to the MML file a file containing simulation frames is loaded. Each frame contains information of the simulation at a specific point in time.

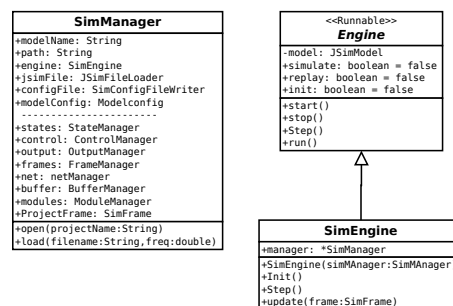


Figure B-8.: Simulation Manager and Simulation Engine

Simulation Frames

A simulation frame consists of control variables, that determine the current values that remain constant through out a simulation step, state variables that are used for initialization of the differential equations in the model, and output variables extracted after the simulation step has been executed. With this frame structure it is possible to recreate different simulation scenarios going through different stages.

Sim file structure

For the creation of files containing the simulation frames XML was used. The following is an example of how these files are structured.

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<AM_SIM version="0.0.1">
  <project name="">
    <model name="MODEL">
      <StateVarsConfig>
        <control group="0" index="0" max="1.0" min="0.0" name="EDPLV" presicion="3" visible="false"/>
        ...
      </StateVarsConfig>
    </model>
  </project>
</AM_SIM>
  
```

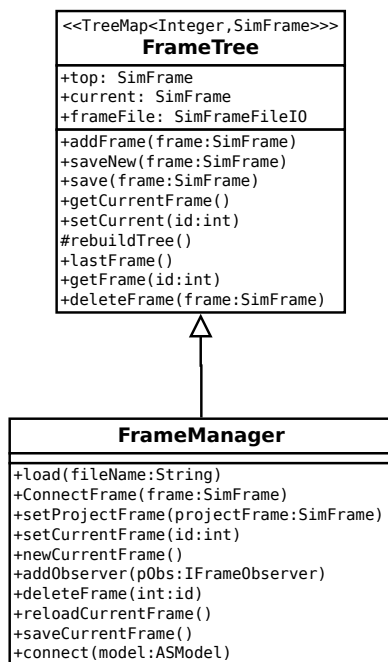


Figure B-9.: Frame contents

```

10     <ControlVarsConfig>
        <control group="0" index="0" max="1.0" min="0.0" name="Vlvrd" presicion="3" visible="false"/>
        ...
    </ControlVarsConfig>
    <OutputVarsConfig>
        <control index="0" module="0" name="Vcorev:t" send="true"/>
        ...
    </OutputVarsConfig>
</model>
<Frames>
    <Frame0>
20     <parameter id="0" parent="-1" type="keyFrame"/>
        <time end="1.0" init="0.0" step="0.1"/>
        <StateVars>
            <control name="COutput" value="108.44352511"/>
            ...
        </StateVars>
        <ControlVars>
            <control name="A" value="-2.17"/>
            ...
        </ControlVars>
30    </Frame0>
    <Frame1>
        ...
    </Frame1>
</Frames>
</project>
</AM_SIM>
    
```

Mathematical Model (MML)

This section contains the mathematical model written in the Mathematical Model Language used in JSim. This is organized in several sections, starting with the patient illnesses and patient parameters, then the cardiovascular model containing the heart and systemic model. Afterwards comes the model of the ECSS, then a baroreceptor model describing the heart activation, then a medication model starting with the volume concentration, then vasoconstrictor and vasodilator, and afterwards the oxygen exchange model and carbon dioxide.

1. Global Variables

```

i>>:// TITLE: Cardiovascular and ECSS model with gas exchange, medication and
//          hematocrit calculation
// CODE LANGUAGE: MML
// CODE:
JSim v1.1
    import nsrunit;
    unit conversion on; // Double slashes indicate a comment
    unit rev = 2*3.1416 radian;
//<GAS EXCHANGE>
10    unit mlSTPD = 1/22400 mole;
//</GAS EXCHANGE>
    math simpleRC3 {
realDomain t sec; t.min=0; t.max=5.0; t.delta=0.1;

//-----
// II.8 VARIABLES OF ILLNESSES
//-----
extern real ILL_FVOL_LOSS(t) ml/min;
// illness: Change in Heart rate for baroreflex regulation
20 extern real ILL_HR 1/min;
// illness: Change in Maximum elastance of first left/right ventricle component
extern real ILL_Emaxlv mmHg/ml,
            ILL_Emaxrv mmHg/ml;
// illness: Change in Mean proximal aortic pressure for baroreflex control
extern real ILL_MAP mmHg;
real HT_E2max mmHg/ml; //Emaxlv
real HT_E4max mmHg/ml; //Emaxrv
HT_E2max = 5.59999999999999964473 * ILL_Emaxlv;
HT_E4max = 0.31700000000000000400 * ILL_Emaxrv;
30 // CONTROL PARAMETERS

```

C. Mathematical Model (MML)

```

extern real MEDS_VOL_IN(t) L/min;
//-----
//  PARAMETERS PATIENT
//-----
extern real PARAM_tweight kg;
extern real PARAM_bsa m^2;
//*****//
//Model General Parameters
private real dens = 1000.00      kg/m^3;      // Blood density
40 private real g = 9.80000000000000071054    m/s^2;      // Gravity
//*****//
real y1(t) dimensionless;//yla    // Activation function for ventricular elastance
real y2(t) dimensionless;//ylv    // Activation function for ventricular elastance
real y3(t) dimensionless;//yra    // Activation function for ventricular elastance
real y4(t) dimensionless;//yrv    // Activation function for ventricular elastance
real Pext = 0.00      mmHg;      // External pressure
real y0 = 0.00      dimensionless; // Cut off for chopped sin function
//real HR = 128.92413979377462851517      hz;      // Heart rate
//<BARORECEPTORS>
50 real      HR(t) 1/min; // Heart rate
//</BARORECEPTORS>

```

2. Cardiovascular System

```

i>>:
/*
  XXXXXXXX  XXXXXXX  XXXX  X  X
  X      X X      X X  X  X  X
  X      X X      X X  X  X X
  XXXXXXXX  X      X X  X  X
  X      X X      X X  X  X
  X      X X      X X  X  X
  XXXXXXXX  XXXXX  XXX  X
10 */
//-----
//HEART MODEL
//-----
real HT_R1 = 0.001      mmHg*sec/ml; //Rla //Mitral Valve resistance
real HT_R2 = 0.001      mmHg*sec/ml;
real HT_R3 = 0.001      mmHg*sec/ml; //Rra //Tricuspid Valve resistance
real HT_R4 = 0.001      mmHg*sec/ml;
real PRint = 0.15      sec; // Difference in atrial, ventricular activation times
20 real HT_E1max = 0.145      mmHg/ml; //Emaxla
//real HT_E2max = 5.6      mmHg/ml; //Emaxlv
real HT_E3max = 0.1940      mmHg/ml; //Emaxra
//real HT_E4max = 0.3170      mmHg/ml; //Emaxrv

real HT_E1min = 0.1940      mmHg/ml; //Eminla
real HT_E2min = 0.187      mmHg/ml; //Eminlv
real HT_E3min = 0.118      mmHg/ml; //Eminra
real HT_E4min = 0.167      mmHg/ml; //Eminrv
//Systolic
real HT_restV1s = 8.00      ml;//Vlars // End-systolic rest volume of ventricle#
30 real HT_restV2s = 23.00      ml;//Vlvrs // End-systolic rest volume of ventricle
real HT_restV3s = 8.00      ml;//Vrars // End-systolic rest volume of ventricle
real HT_restV4s = 53.00      ml;//Vrvrs // End-systolic rest volume of ventricle
//Diastolic
real HT_restV1d = 10.00      ml;//Vlard // End-diastolic rest volume of ventricle
real HT_restV2d = 72.00      ml;//Vlvrd // End-diastolic rest volume of ventricle
real HT_restV3d = 10.00      ml;//Vrard // End-diastolic rest volume of ventricle
real HT_restV4d = 53.00      ml;//Vrvrs // End-diastolic rest volume of ventricle
//++Variables
//Pressure

```

```

40 real HT_P1(t) mmHg;
real HT_P2(t) mmHg;
real HT_P3(t) mmHg;
real HT_P4(t) mmHg;
//Volume
real HT_V1(t) ml;
real HT_V2(t) ml;
real HT_V3(t) ml;
real HT_V4(t) ml;
//Elastance
50 real HT_E1(t) mmHg/ml;
real HT_E2(t) mmHg/ml;
real HT_E3(t) mmHg/ml;
real HT_E4(t) mmHg/ml;
//Flow
real HT_F1(t) ml/s;
real HT_F2(t) ml/s;
real HT_F3(t) ml/s;
real HT_F4(t) ml/s;
//Rest volume
60 real HT_restV1(t) ml;           // Rest volume of ventricle
real HT_restV2(t) ml;           // Rest volume of ventricle
real HT_restV3(t) ml;           // Rest volume of ventricle
real HT_restV4(t) ml;           // Rest volume of ventricle

//Heart rate calculation
private real TsvK = 0.3 sec; // Scaler to set ventricular systolic fraction of heart cycle
private real TsaK = 0.2 sec; // Scaler to set atrial systolic fraction of heart cycle

realState Tsv(t) sec; // Discrete systolic interval for ventricles
70 realState Tsa(t) sec; // Discrete systolic interval for atria

real trela(t) sec;
real trelv(t) sec;

real Tsa__init(t);
real Tsv__init(t);

real tshift__init = 0;

80 realState tshift(t) sec; // Time offset for heart rate
when(t=t.min) {
    tshift=tshift__init;

    Tsa__init=TsaK*sqrt(1.0/HR/(1.0 sec));
    Tsv__init=TsvK*sqrt(1.0/HR/(1.0 sec));

    Tsv = Tsv__init;
    Tsa = Tsa__init;
}

90 event (t+t.delta>=(tshift+(1/HR))){
    Tsv = TsvK*sqrt(1.0/HR/(1.0 sec));
    Tsa = TsaK*sqrt(1.0/HR/(1.0 sec));
    tshift = t;
}
//-----
// HEART ACTIVATION
//-----
//<CONT_SIM>
100 Tsa__init:t =0;
Tsv__init:t=0;
//</CONT_SIM>
//y = sin(2*PI*HR*t); // Activation function

```

C. Mathematical Model (MML)

```
trela = t-tshift;
trelv = t-tshift-Print;

y1 = if (0.0<=trela and trela<Tsa)
      (1.0 - cos(PI*trela/Tsa))/2.0
110   else if (Tsa<=trela and trela<1.5*Tsa)
      (1.0 + cos(2.0*PI*(trela-Tsa)/Tsa))/2.0
      else
      0.0;
y3 = y1;
y2 = if (0.0<=trelv and trelv<Tsv)
      (1.0 - cos(PI*trelv/Tsv))/2.0
      else if (Tsv<=trelv and trelv<1.5*Tsv)
      (1.0 + cos(2.0*PI*(trelv-Tsv)/Tsv))/2.0
120   else
      0.0;
y4 = y2;
//-----
//SYSTEMIC MODEL
//-----

//++Head circulation
//Resistances
real SUP2_R1 = 7.43499999999999960920      mmHg*sec/ml;
real SUP2_F1(t) ml/s;
130

//++Upper Systemic
//Resistances
real SYS_R0 = 0.01000000000000000021      mmHg*sec/ml;
real SYS_R0 = 0.01000000000000000021      mmHg*sec/ml;
//real SYS_R1 = 1.00      mmHg*sec/ml;
//EFFECT
real SYS_R1_0 = 1.00      mmHg*sec/ml;
real SYS_R1(t) mmHg*sec/ml;
140

real SYS_R2 = 0.03200000000000000067      mmHg*sec/ml;
real SYS_R3 = 0.048000000000000000100      mmHg*sec/ml;
real SYS_R4 = 0.11799999999999999378      mmHg*sec/ml;
real SYS_R5 = 0.081000000000000000255      mmHg*sec/ml;

//Compliances
real SYS_C0 = 0.042000000000000000261      ml/mmHg;
real SYS_C0 = 0.06300000000000000044      ml/mmHg;
//real SYS_C1 = 3.51600000000000001421      ml/mmHg;
150 real SYS_C1_0 = 3.51600000000000001421      ml/mmHg;
real SYS_C1(t) ml/mmHg;
real SYS_C2 = 10.00      ml/mmHg;
//real SYS_C3 = 40 ml/mmHg;
real SYS_C4 = 4.65099999999999980105      ml/mmHg;
real SYS_C5 = 4.65099999999999980105      ml/mmHg;

//Variables
//Flows
real SYS_F00(t) ml/s;
160 real SYS_F0(t) ml/s;
real SYS_F1(t) ml/s;
real SYS_F2(t) ml/s;
real SYS_F3(t) ml/s;
real SYS_F4(t) ml/s;
real SYS_F5(t) ml/s;
//Pressures
real SYS_P00(t) mmHg;
real SYS_P0(t) mmHg;
real SYS_P1(t) mmHg;
```



```

170 real SYS_P2(t) mmHg;
    real SYS_P3(t) mmHg;
    real SYS_P4(t) mmHg;
    real SYS_P5(t) mmHg;
    //Volumes
    real SYS_V00(t) ml;
    real SYS_V0(t) ml;
    real SYS_V1(t) ml;
    real SYS_V2(t) ml;
    real SYS_V3(t) ml;
180 real SYS_V4(t) ml;
    real SYS_V5(t) ml;
    //Resting Volumes
    real SYS_restV00 = 5.00      ml;
    real SYS_restV0 = 5.00      ml;
    real SYS_restV1 = 453.08    ml;
    real SYS_restV2 = 85.453    ml;
    real SYS_restV4 = 0.134     ml;
    real SYS_restV5 = 0.091     ml;

190 private real Kv = 20.00      mmHg; // Scaling factor for systemic venous pressure
    real SYS_V3max = 3000.00    ml; // Maximal volume of lumped systemic veins
    //++Upper systemic

    real SUP_V3max = 480.00     ml; // Maximal volume of lumped systemic veins
    //Resistances
    real SUP_R0 = 0.10000000000000000555 mmHg*sec/ml;
    real SUP_R1 = 0.50000000000000000000 mmHg*sec/ml;
    real SUP_R2 = 0.2469999999999999734 mmHg*sec/ml;
    real SUP_R3 = 0.37600000000000000089 mmHg*sec/ml;
200 //Compliances
    real SUP_C1 = 1.00          ml/mmHg;
    real SUP_C2 = 0.558000000000000005151 ml/mmHg;
    //real SUP_C3 = 40 ml/mmHg;

    //++Variables
    //Flow
    real SUP_F0(t) ml/s;
    real SUP_F1(t) ml/s;
    real SUP_F2(t) ml/s;
210 real SUP_F3(t) ml/s;
    //Pressure
    real SUP_P1(t) mmHg;
    real SUP_P2(t) mmHg;
    real SUP_P3(t) mmHg;
    //Volume
    real SUP_V1(t) ml;
    real SUP_V2(t) ml;
    real SUP_V3(t) ml;
    //Rest Volumes
220 real SUP_restV1 = 30.00      ml;
    real SUP_restV2 = 10.00     ml;
    //-----
    //PULMONARY MODEL
    //-----
    real PM_R1 = 0.01000000000000000021 mmHg*sec/ml;
    real PM_C1 = 1.35699999999999998401 ml/mmHg;
    real PM_F1(t) ml/s;
    real PM_P1(t) mmHg;
    real PM_V1(t) ml;
230 real PM_restV1 = 377.41899999999998271960 ml;

```

3. ECSS Model

```

i>>ε//*****//
/*
  XXXXXXXXX  XXXXXX  XXXXXX  XXXXX
  X          X      X      X
  X          X      X      X
  XXXXX     X      XXXXXX  XXXXX
  X          X          X      X
  X          X          X      X
  XXXXXXXXX  XXXXXX  XXXXXX  XXXXXX
10 */
//CLAMPS
extern real HLM_ClampV mmHg*sec/ml;
real cHLM_ClampV(t) mmHg*sec/ml;
extern real HLM_ClampA mmHg*sec/ml;
real cHLM_ClampA(t) mmHg*sec/ml;
extern real HLM_ClampSpeed;
//Clamp calculation for opening and closing
  cHLM_ClampV:t= if (HLM_ClampV*1000>cHLM_ClampV)
20     HLM_ClampSpeed*(HLM_ClampV*1000-cHLM_ClampV) *(1 1/sec)
     else if (cHLM_ClampV>HLM_ClampV*1000)
        -HLM_ClampSpeed*(cHLM_ClampV-HLM_ClampV*1000) *(1 1/sec)
     else 0;

  cHLM_ClampA:t= if (HLM_ClampA*1000>cHLM_ClampA)
        HLM_ClampSpeed*(HLM_ClampA*1000-cHLM_ClampA) *(1 1/sec)
     else if (cHLM_ClampA>HLM_ClampA*1000)
        -HLM_ClampSpeed*(cHLM_ClampA-HLM_ClampA*1000) *(1 1/sec)
     else 0;
30 when (t=t.min){
  cHLM_ClampV=1000;
  cHLM_ClampA=1000;
}

//Resistances
real HLM_RIn(t) mmHg*sec/ml;
real HLM_R01 = 1.00 mmHg*sec/ml;
real HLM_R02 = 1.00 mmHg*sec/ml;
real HLM_R1 = 1.00 mmHg*sec/ml;
40 real HLM_R4 = 1.00 mmHg*sec/ml;
real HLM_R5 = 1.00 mmHg*sec/ml;
real HLM_R7 = 1.00 mmHg*sec/ml;
real HLM_R8 = 1.00 mmHg*sec/ml;
real HLM_ROut(t) mmHg*sec/ml;
//Flows
real HLM_F0(t) L/min;
real HLM_F1(t) L/min;
real HLM_F4(t) L/min;
//Pressure
50 real HLM_P01(t) mmHg;
real HLM_P02(t) mmHg;
real HLM_P1(t) mmHg;
real HLM_P2(t) mmHg;
real HLM_P3(t) mmHg;
real HLM_P4(t) mmHg;
real HLM_P5(t) mmHg;
real HLM_P6(t) mmHg;
real HLM_P7(t) mmHg;
real HLM_P8(t) mmHg;
60 //Volumes
real HLM_V1(t) ml;
real HLM_V4(t) ml;
//Rest volumes

```

```

real HLM_restV1 = 500.00      ml;
real HLM_restV4 = 280.00      ml;
//Compliances
real HLM_C1 = 1.00           ml/mmHg;
real HLM_C4 = 0.0100000000000000021      ml/mmHg;
//Inductance
70 real HLM_L1 = 1.00           mmHg*sec^2*ml^(-1);

//Formulas
HLM_RIn = HLM_R01 + HLM_R02 + cHLM_ClampV;
HLM_ROut = HLM_R4 + HLM_R5 + cHLM_ClampA + HLM_R7 + HLM_R8;

HLM_P01 = SYS_P4 - HLM_F0 * HLM_R01;
HLM_P02 = HLM_P01 - HLM_F0 * HLM_R02;

HLM_P5 = HLM_P4 - HLM_F4 * HLM_R4;
80 HLM_P6 = HLM_P5 - HLM_F4 * HLM_R5;
HLM_P7 = HLM_P6 - HLM_F4 * cHLM_ClampA;
HLM_P8 = HLM_P7 - HLM_F4 * HLM_R7;
    real DeltaP_Pump(t) mmHg;

//#COMPONENT#    PUMP{
//*****//
//                PUMP
//*****//
//*****INPUTS*****//
90  extern real InputSpeed(t) rev/min;
//*****CONSTANTS*****//
//CONSTANTS//
    real area m^2;           // Output pump area
private real r = 0.00500000000000000010      m; // Pump output radius
    area=PI*r^2;
private real PumpAcc = 237.00 rev/min/sec; //Defines the speed increase for every second

//Constants for the head
private    real Pump_C1 = 0.00      ;
100 private    real Pump_C2 = 0.02199999999999999872      ;
private    real Pump_C3 = 0.002000000000000000004      m*min/rev;
private real Pump_C4 = -2.00      m;

//*****VARIABLES*****//
    real pumpDirection(t) 1/sec;
        //used to calculate the correct acc
    real cPumpSpeed(t) rev/min;
    real Q_Pump(t) L/min;
    real Pump_s(t); //Pump Static
110    real Pump_d(t); //Pump Dynamic
//*****EQUATIONS*****//
    pumpDirection=if(InputSpeed>cPumpSpeed)
        1
        else
            -1;
//Pump speed defined by PumpAcceleration
    cPumpSpeed:t= if(cPumpSpeed=InputSpeed) 0
        else
            if(abs(InputSpeed-cPumpSpeed)<PumpAcc*(1 sec))
120                abs(InputSpeed-cPumpSpeed)*pumpDirection
            else
                if(InputSpeed>cPumpSpeed)
                    PumpAcc
                else
                    -PumpAcc;
private real conv = 1.00      min^2*m/(L*rev);

//Constants for Q out of H(Q);

```

C. Mathematical Model (MML)

```
//Pump_dyn slope of H over RPM, Pump_static Constant of H over RPM
130      Pump_d=(Pump_C1*cPumpSpeed+Pump_C2*(1 rev/min))*conv;
          Pump_s=if (cPumpSpeed<1000) 0
                else Pump_C3*cPumpSpeed+Pump_C4 ;
      when(t=t.min){
        cPumpSpeed = 1000.00      ;
      }
      DeltaP_Pump=(Pump_s+Pump_d*Q_Pump)*dens*g;
//#COMPONENT END#      }

140 Q_Pump=HLM_F1;
//-----
real ctPlasma_SYS_3(t);
extern real MEDS_kVL;
```

4. Equations

```
i>>
//-----
//EQUATIONS
//-----

HT_E1 = (HT_E1max-HT_E1min)*y1 + HT_E1min; // Elastance driver from activation fxn
HT_E2 = (HT_E2max-HT_E2min)*y2 + HT_E2min; // Elastance driver from activation fxn
HT_E3 = (HT_E3max-HT_E3min)*y3 + HT_E3min; // Elastance driver from activation fxn
HT_E4 = (HT_E4max-HT_E4min)*y4 + HT_E4min; // Elastance driver from activation fxn

10 HT_restV1 = (1-y1)*(HT_restV1d-HT_restV1s) + HT_restV1s;
   HT_restV2 = (1-y2)*(HT_restV2d-HT_restV2s) + HT_restV2s;
   HT_restV3 = (1-y3)*(HT_restV3d-HT_restV3s) + HT_restV3s;
   HT_restV4 = (1-y4)*(HT_restV4d-HT_restV4s) + HT_restV4s;

private real Px2 = 2.00      mmHg;      // P-V curve shaping parameter
private real Vx8 = 8.00      ml;        // P-V curve shaping parameter

HT_P1 = (HT_V1-HT_restV1)*HT_E1
20   - Px2*(1/(exp((HT_V1)/Vx8)-1)); // Eq. B,F
HT_P2 = (HT_V2-HT_restV2)*HT_E2
   - Px2*(1/(exp((HT_V2)/Vx8)-1)); // Contractility effects systolic elastance
HT_P3 = (HT_V3-HT_restV3)*HT_E3
   - Px2*(1/(exp((HT_V3)/Vx8)-1)); // Eq. B,F
HT_P4 = (HT_V4-HT_restV4)*HT_E4
   - Px2*(1/(exp((HT_V4)/Vx8)-1)); // Eq. B,F

SYS_P00 = (SYS_V00-SYS_restV00)/SYS_C00 + Pext;
SYS_P0 = (SYS_V0-SYS_restV0)/SYS_C0 + Pext;
30 SYS_P1 = if (SYS_V1>=SYS_restV1) ((SYS_V1-SYS_restV1)/SYS_C1) + Pext // Eq. B
   else (-SYS_restV1/SYS_C1) + Pext; // no negative volumes in equations
SYS_P2 = if (SYS_V2>=SYS_restV2) ((SYS_V2-SYS_restV2)/SYS_C2) + Pext // Eq. B
   else (-SYS_restV2/SYS_C2) + Pext; // no negative volumes in equations
SYS_P3=-Kv*log((SYS_V3max/SYS_V3)-0.99); // Lu et al. Eq. (1)
//-----
SUP_P1 = if (SUP_V1>=0) ((SUP_V1-SUP_restV1)/SUP_C1) + Pext // Eq. B
   else (-SUP_restV1/SUP_C1) + Pext; // no negative volumes in equations
SUP_P2 = if (SUP_V2>=0) ((SUP_V2-SUP_restV2)/SUP_C2) + Pext // Eq. B
   else (-SUP_restV2/SUP_C2) + Pext; // no negative volumes in equations
40 SUP_P3=-Kv*log((SUP_V3max/SUP_V3)-0.99); // Lu et al. Eq. (1)
PM_P1 = if (PM_V1>=PM_restV1) ((PM_V1-PM_restV1)/PM_C1) + Pext // Eq. B
   else (-PM_restV1/PM_C1) + Pext; // no negative volumes in equations
SYS_P4 = if (SYS_V4>=SYS_restV4) ((SYS_V4-SYS_restV4)/SYS_C4) + Pext // Eq. B
```

```

        else (-SYS_restV4/SYS_C4) + Pext;          // no negative volumes in equations
SYS_P5 = if (SYS_V5>=SYS_restV5) ((SYS_V5-SYS_restV5)/SYS_C5) + Pext // Eq. B
        else (-SYS_restV5/SYS_C5) + Pext;          // no negative volumes in equations
//-----
SYS_F00 = (SYS_P00-SYS_P0)/SYS_R00;
SYS_F0 = (SYS_P0-SYS_P1)/SYS_R0;
50 SYS_F1 = (SYS_P1-SYS_P2)/SYS_R1;
SYS_F2 = (SYS_P2-SYS_P3)/SYS_R2;
SYS_F3 = (SYS_P3-SYS_P4)/SYS_R3;
SYS_F4 = (SYS_P4-SYS_P5)/SYS_R4;
SYS_F5 = (SYS_P5-HT_P3)/SYS_R5;

SUP2_F1 = (SYS_P00 - SYS_P5) / SUP2_R1;

SUP_F0 = (SYS_P00-SUP_P1)/SUP_R0;
SUP_F1 = (SUP_P1-SUP_P2)/SUP_R1;
60 SUP_F2 = (SUP_P2-SUP_P3)/SUP_R2;
SUP_F3 = (SUP_P3-SYS_P5)/SUP_R3;
//ILLNESS-----
real ILL_G_mit_back=0 ml/mmHg/sec;

HT_F1 = if (HT_P1>HT_P2) (HT_P1-HT_P2)/ HT_R1
//      else 0;
        else (HT_P1-HT_P2)*ILL_G_mit_back;
//Heart flow-----
HT_F2 = if(HT_P2>SYS_P00-0.2) (HT_P2-SYS_P00)/HT_R2
70     else 0;
HT_F3 = if(HT_P3>HT_P4) (HT_P3-HT_P4)/HT_R3
        else 0;
HT_F4 = if(HT_P4>PM_P1) (HT_P4-PM_P1) / HT_R4
        else 0;
PM_F1 = (PM_P1-HT_P1) / PM_R1;
HLM_F0 = (SYS_P4-HLM_P1)/HLM_RIn;
//Pressure calculation-----
HLM_P1= (HLM_V1-HLM_restV1)/HLM_C1;
HLM_P4= (HLM_V4-HLM_restV4)/HLM_C4;
80 HLM_P3=HLM_P2+DeltaP_Pump;
//Flow calculation-----
HLM_F1= (HLM_P1-HLM_P2)/HLM_R1;
HLM_F1:t= (HLM_P3-HLM_P4)/HLM_L1; //CHECK SIGN!!
HLM_F4= if (HLM_P4>SYS_P0) (HLM_P4-SYS_P0)/HLM_ROut
        else
        0;
//Volume calculation -----
HT_V1:t= PM_F1 - HT_F1;
HT_V2:t= HT_F1 - HT_F2;
90 HT_V3:t= SYS_F5 - HT_F3;
HT_V4:t= HT_F3 - HT_F4;

PM_V1:t = HT_F4 - PM_F1;

SYS_V00:t = HT_F2 - SYS_F00 - SUP_F0 - SUP2_F1;
SYS_V0:t = SYS_F00 - SYS_F0 + HLM_F4;

SYS_V1:t = SYS_F0 - SYS_F1;
SYS_V2:t = SYS_F1 - SYS_F2;
100 SYS_V3:t = SYS_F2 - SYS_F3 - ILL_FVOL_LOSS - MEDS_kVL*ctPlasma_SYS_3;

SYS_V4:t = SYS_F3 - SYS_F4 - HLM_F0;
SYS_V5:t = SYS_F4 + SUP2_F1 + SUP_F3 - SYS_F5;

SUP_V1:t = SUP_F0 - SUP_F1;
SUP_V2:t = SUP_F1 - SUP_F2;
SUP_V3:t = SUP_F2 - SUP_F3;

```

C. Mathematical Model (MML)

```
HLM_V1:t = HLM_F0 - HLM_F1 + MEDS_VOL_IN;
110 HLM_V4:t = HLM_F1 - HLM_F4;

when (t=t.min) { //Initial conditions
    HLM_F1 = -0.00031102959999999998 ;
    HT_V1 = 20.089689209999999951960 ;
    HT_V2 = 53.62983425000000181626 ;
    HT_V3 = 28.771890979999999845001 ;
    HT_V4 = 60.70588224000000110436 ;
    SYS_V00 = 7.93874010000000041032 ;
    SYS_V0 = 9.16648879999999977031 ;
120    SYS_V1 = 672.21370999999999185093 ;
    SYS_V2 = 266.99732999999997673513 ;
    SYS_V3 = 2643.6550000000020008883 ;
    SYS_V4 = 8.32938251586997857601 ;
    SYS_V5 = 8.32938251586997857601 ;
    SUP_V1 = 88.382384490000000684919 ;
    SUP_V2 = 30.1352900000000124260 ;
    SUP_V3 = 461.01594999999997526174 ;
    PM_V1 = 380.24707999999998264684 ;
130    HLM_V1 = 507.6316400000000442014 ;
    HLM_V4 = 280.07632000000000971340 ;
}
//Total volume calculation-----
real V_TOTAL(t) ml;
real V_BODY(t) ml;

V_BODY = HT_V1+HT_V2+HT_V3+HT_V4+SYS_V00+SYS_V0+SYS_V1+SYS_V2+
        SYS_V3+SYS_V4+SYS_V5+SUP_V1+SUP_V2+SUP_V3+PM_V1;
V_TOTAL = V_BODY + +HLM_V1+HLM_V4;

140 // Derived variables-----
real COutput(t) ml*sec-1; // Cardiac output
real CO(t) L/min; // Cardiac output
real EFR(t) L/min; // Cardiac output
real MAP(t) mmHg;
real CVP(t) mmHg;

real SVR(t) dyn*s/cm5;
private real COTau = 10.00 sec; // Cardiac output equation time constant
COutput:t = (HT_F2-COutput)/COTau; // Cardiac output
150 CO=COutput;

MAP:t = (SYS_P00-MAP)/COTau;
EFR:t = (HLM_F4-EFR)/COTau;
CVP:t = (HT_P3-CVP)/COTau;
SVR = (MAP-CVP)/(HLM_F4);
//SVR = (MAP-CVP)/(HLM_F4+CO);

when(t=t.min){
160    COutput = 94.23590145999999379001 ;
    MAP = 62.5484200000000012960 ;
    CVP = 1.94435160000000006875 ;
    EFR = 1;
}

real HRBase = 100.00 1/min; // Heart rate
//</BARORECEPTORS>
```

5. Baroreceptor

```
i>>i
//-----
```

```

/*
  BARORECEPTOR
*/
//-----
// I.4 PARAMETERS OF BARORECEPTOR MODEL
//-----
10 private real baroC = 0.063;           // Factor for baroreflex function curve
   real baroRefMAP = 100.00 mmHg;      // Reference MAP for baroreflex
   private real baroRes_Gain = -1.419999999999992895 ;
   private real baroRes_Delay = 3.00 sec;
   private real baroRes_Tau = 6.00 sec;
   private real baroEm_Gain = -0.479999999999998224 ;
   private real baroEm_Delay = 3.00 sec;
   private real baroEm_Tau = 10.00 sec;
   private real baroHr_Gain = -0.832999999999996270 ;
20 private real baroHr_Delay = 0.2000000000000001110 sec;
   private real baroHr_Tau = 1.8000000000000004441 sec;
   private real baroVvol_Gain = 250.00 ;
   private real baroVvol_Delay = 10.00 sec;
   private real baroVvol_Tau = 60.00 sec;
   private real factorMin = 1.00 1/min;

//-----
// II.4 VARIABLES OF BARORECEPTOR MODEL
//-----
30 private real bfc(t); // baroreflex function curve
   private real baroRes_State1(t),
      baroRes_State2(t),
      baroEm_State1(t),
      baroEm_State2(t),
      baroHr_State1(t),
      baroHr_State2(t),
      baroVvol_State1(t),
      baroVvol_State2(t);

40 real baroTStep(t);
   real baroActy(t);

   when (t=t.min){
     baroActy = 0.08754016000000000575 ;
     baroRes_State1 = 0.08754741000000000606 ;
     baroRes_State2 = -0.12435485000000000289 ;
     baroEm_State1 = 0.08754741000000000606 ;
     baroEm_State2 = -0.04213842999999999756 ;
     baroHr_State1 = 0.08754246999999999723 ;
50 baroHr_State2 = -0.07292524999999999702 ;
     baroVvol_State1 = 0.08771610000000000518 ;
     baroVvol_State2 = 28.80626197999999860144 ;
   }
//-----
// III.4 EQUATIONS OF BARORECEPTOR MODEL
//-----

   bfc = exp(baroC * (MAP-(baroRefMAP+ILL_MAP)))/(1.0+exp(baroC * (MAP-(baroRefMAP+ILL_MAP))));
   //baroTStep = (60.0 * factorMin) / (HR * 180); // TODO t.delta?
60 baroTStep = HR / (180 * factorMin);
   baroActy:t = baroTStep * (1.0*bfc - baroActy)/(4.0 sec);
   baroRes_State1:t = baroTStep * (1.0*baroActy - baroRes_State1)/baroRes_Delay;
   baroRes_State2:t = baroTStep * (baroRes_Gain*baroRes_State1 - baroRes_State2)/baroRes_Tau;
   baroEm_State1:t = baroTStep * (1.0*baroActy - baroEm_State1)/baroEm_Delay;
   baroEm_State2:t = baroTStep * (baroEm_Gain*baroEm_State1 - baroEm_State2)/baroEm_Tau;
   baroHr_State1:t = baroTStep * (1.0*baroActy - baroHr_State1)/baroHr_Delay;
   baroHr_State2:t = baroTStep * (baroHr_Gain*baroHr_State1 - baroHr_State2)/baroHr_Tau;

```

C. Mathematical Model (MML)

```
baroVvol_State1:t = baroTStep *(1.0*baroActy - baroVvol_State1)/baroVvol_Delay;
baroVvol_State2:t = baroTStep *(baroVvol_Gain*baroVvol_State1 - baroVvol_State2)/baroVvol_Tau;
70
//Emaxlv = (1.0 - 0.5*baroEm_Gain + baroEm_State2) * (EmaxlvBase + illEmaxlv);
//Emaxrv = (1.0 - 0.5*baroEm_Gain + baroEm_State2) * (EmaxrvBase + illEmaxrv);
//Rsa = (1.0 - 0.5*baroRes_Gain + baroRes_State2) * (RsaBase);
HR = (1.0 - 0.5*baroHr_Gain + baroHr_State2) * (HRBase + ILL_HR);
```

6. Gas Exchange

```
i>>:
private real Vcvttox = 2.1e-7 mol/min/g; // Vmax for CYTOCHROME OXIDASE in SYST ISF
private real Kcvttox = 1.0e-9 M; // Km for CYTOCHROME OXIDASE in SYST ISF
//Saturations
private real nH = 3.50000000000000000000000000000000 dimensionless; //Hill coefficient
private real P50_O2 = 26.50000000000000000000000000000000 mmHg; //P50 of O2 for Hb
//Concentrations
private real CHb = 0.0204 M; // Concentration of O2 or CO2 binding sites on Hb
//real Hcrit = 0.45 dimensionless; // Hematocrit
10 real Hcrit(t);
real cHb(t) g/L;
Hcrit=hct_SYS_3*(1 dL/g);
cHb=Hcrit/3*(1 g/dL);
real GAS_alphaO2 = 1.36e-6 M/mmHg; // Solubility constant of O2 in plasma
// (assume that is is the same for red blood cells)
real SolOxy =0.031 mL/mmHg/L;
real HbO2Content = 1.34 mL/g;
//VARIABLES
//Partial Pressures
20 real PO2_SYS_1(t) ;
real PO2_SYS_2(t) ;
real PO2_SYS_3(t) ;
real PO2_PM_1(t) ;
real PO2_HLM_1(t) ;
real PO2_HLM_4(t) ;
//Concentrations
real CtO2_SYS_1(t) mL/L;
real CtO2_SYS_2(t) mL/L;
real CtO2_SYS_3(t) mL/L;
30 real CtO2_PM_1(t) mL/L;
real CtO2_HLM_1(t) mL/L;
real CtO2_HLM_4(t) mL/L;
//Saturations
real SHbO2_SYS_1(t) ;
real SHbO2_SYS_2(t) ;
real SHbO2_SYS_3(t) ;
real SHbO2_PM_1(t) ;
real SHbO2_HLM_1(t) ;
real SHbO2_HLM_4(t) ;
40 real dSHbO2_SYS_1(t) ;
real dSHbO2_SYS_2(t) ;
real dSHbO2_SYS_3(t) ;
real dSHbO2_PM_1(t) ;
real dSHbO2_HLM_1(t) ;
real dSHbO2_HLM_4(t) ;
//Calculations
SHbO2_SYS_1= (PO2_SYS_1/P50_O2)^nH/(((PO2_SYS_1/P50_O2)^nH) + 1)*100;
SHbO2_SYS_2= (PO2_SYS_2/P50_O2)^nH/(((PO2_SYS_2/P50_O2)^nH) + 1)*100;
SHbO2_SYS_3= (PO2_SYS_3/P50_O2)^nH/(((PO2_SYS_3/P50_O2)^nH) + 1)*100;
```



```

50   SHbO2_PM_1= (PO2_PM_1/P50_O2)^nH/(((PO2_PM_1/P50_O2)^nH) + 1)*100;
      SHbO2_HLM_1= (PO2_HLM_1/P50_O2)^nH/(((PO2_HLM_1/P50_O2)^nH) + 1)*100;
      SHbO2_HLM_4= (PO2_HLM_4/P50_O2)^nH/(((PO2_HLM_4/P50_O2)^nH) + 1)*100;
//   dSHb/dP
      dSHbO2_SYS_1 =nH*(PO2_SYS_1 ^-1)*(PO2_SYS_1 /P50_O2)^nH/(((PO2_SYS_1 /P50_O2)^nH) + 1)^2;
      dSHbO2_SYS_2 =nH*(PO2_SYS_2 ^-1)*(PO2_SYS_2 /P50_O2)^nH/(((PO2_SYS_2 /P50_O2)^nH) + 1)^2;
      dSHbO2_SYS_3 =nH*(PO2_SYS_3 ^-1)*(PO2_SYS_3 /P50_O2)^nH/(((PO2_SYS_3 /P50_O2)^nH) + 1)^2;
      dSHbO2_PM_1 =nH*(PO2_PM_1 ^-1)*(PO2_PM_1 /P50_O2)^nH/(((PO2_PM_1 /P50_O2)^nH) + 1)^2;
      dSHbO2_HLM_1 =nH*(PO2_HLM_1 ^-1)*(PO2_HLM_1 /P50_O2)^nH/(((PO2_HLM_1 /P50_O2)^nH) + 1)^2;
      dSHbO2_HLM_4 =nH*(PO2_HLM_4 ^-1)*(PO2_HLM_4 /P50_O2)^nH/(((PO2_HLM_4 /P50_O2)^nH) + 1)^2;
60   //CtO2_SYS_1=GAS_alphaO2*PO2_SYS_1+CHb*SHbO2_SYS_1*Hcrit;
      //CtO2_SYS_2=GAS_alphaO2*PO2_SYS_2+CHb*SHbO2_SYS_2*Hcrit;
      //CtO2_SYS_3=GAS_alphaO2*PO2_SYS_3+CHb*SHbO2_SYS_3*Hcrit;
      //CtO2_PM_1=GAS_alphaO2*PO2_PM_1+CHb*SHbO2_PM_1*Hcrit;
      //CtO2_HLM_4=GAS_alphaO2*PO2_HLM_4+CHb*SHbO2_HLM_4*Hcrit;
      CtO2_SYS_1=(HbO2Content * cHb * SHbO2_SYS_1) + SolOxy * PO2_SYS_1;
      CtO2_SYS_2=(HbO2Content * cHb * SHbO2_SYS_2) + SolOxy * PO2_SYS_2;
      CtO2_SYS_3=(HbO2Content * cHb * SHbO2_SYS_3) + SolOxy * PO2_SYS_3;
      CtO2_PM_1=(HbO2Content * cHb * SHbO2_PM_1) + SolOxy * PO2_PM_1;
      CtO2_HLM_1=(HbO2Content * cHb * SHbO2_HLM_1) + SolOxy * PO2_HLM_1;
70   CtO2_HLM_4=(HbO2Content * cHb * SHbO2_HLM_4) + SolOxy * PO2_HLM_4;

      PO2_SYS_1:t = (COutput*(CtO2_PM_1-CtO2_SYS_1)+HLM_F4*(CtO2_HLM_4-CtO2_SYS_1))
                   /((SYS_V1+SUP_V1)*(SolOxy+HbO2Content * cHb *dSHbO2_SYS_1));
      PO2_SYS_3:t = ((COutput+HLM_F4)*(CtO2_SYS_2-CtO2_SYS_3))/
                   ((SYS_V3+SUP_V3)*(SolOxy+HbO2Content * cHb *dSHbO2_SYS_3));
      PO2_PM_1:t = ((COutput)*(CtO2_SYS_3-CtO2_PM_1))/
                   ((PM_V1)*(SolOxy+HbO2Content * cHb *dSHbO2_PM_1));
      PO2_HLM_1:t = ((HLM_F4)*(CtO2_SYS_3-CtO2_HLM_1))/
                   ((HLM_V1)*(SolOxy+HbO2Content * cHb *dSHbO2_HLM_1));
80   when(t = t.min){
      PO2_SYS_1 = 37.58992563999999703128 ;
      PO2_SYS_2 = 37.592074879999999841577 ;
      PO2_SYS_3 = 37.614689429999999853782 ;
      PO2_PM_1 = 37.617456900000000053033 ;
      PO2_HLM_1 = 608.00 ;
      PO2_HLM_4 = 608.00 ;
    }
//Consumption of Oxygen
90   extern real GAS_O2_CONS dimensionless;
      extern real GAS_CO2_PROD dimensionless;
      private real PS = 10.00 ml/min/g; // Permeability-surface area product for gas exchange
      real CtO2_isf(t) M; // Needed for GO2isf - added by mneal
      real PO2_isf(t) mmHg; // O2 partial pressure in interstitial fluid
      real V_O2(t) mole/min; // O2 consumption
//Interstitial fluid
      real Visf = 48000.00 ml; // Value from o2_co2_v1.v11 model by J. Bassingthwaigthe
      CtO2_isf = GAS_alphaO2*PO2_isf;
      V_O2 = if (PO2_isf<=0) 0
100     else (Vcytox*PARAM_tweight*GAS_alphaO2*PO2_isf)/
           (Kcytox + (GAS_alphaO2*PO2_isf)); // O2 consumption
      PO2_isf:t = PS*PARAM_tweight*(PO2_SYS_2-PO2_isf)/Visf - V_O2/(GAS_alphaO2*Visf);
      PO2_SYS_2:t = (COutput+HLM_F4)*(CtO2_SYS_1-CtO2_SYS_2)/
                   ((SYS_V2+SUP_V2)*(SolOxy+HbO2Content * cHb *dSHbO2_SYS_2)) +
                   (PS*PARAM_tweight*Soloxy*(PO2_isf-PO2_SYS_2)*GAS_O2_CONS)/
                   ((SYS_V2+SUP_V2)*(SolOxy+HbO2Content * cHb *dSHbO2_SYS_2));

      when(t=t.min) {
      PO2_isf = 22.154686290000000078122 ;
      }
110 //-----
// X.C BLENDER
//-----
      real Bl_FiO2 = 80.00 dimensionless; //Fractional Inspired Oxygen Concentration

```

C. Mathematical Model (MML)

```
    real Bl_Fgas = 6.00    L/min;           //Gas input flow from blender
//-----
// X.C PARAMETERS OXYGENATOR
//-----
    real Oxy_Vgas = 0.1406000000000000275    L;
120    //Gas volume in Oxygenator (Gunnar p.181)
    //change to own oxigenator volume
    real Oxy_Pref = 760.00    mmHg;
    real Oxy_DO2 = 0.4799999999999998224    mL/(min*mmHg);    // Diffusion factor of Oxygen
    real Oxy_DCO2 = 0.80000000000000004441    mL/(min*mmHg);
    // Diffusion factor of CO2 in oxygenator CHECK!!!
//-----
// X.C VARIABLES OXYGENATOR
//-----
//-----
130 //          OXYGEN
//-----
    real Oxy_PO2i(t) mmHg; // Oxygen input partial pressure in Oxygenator
    real PO2_Oxy_Gas(t) mmHg;    // Oxygen partial pressure in Oxygenator
    real Oxy_K1_O2 = 1;
    real Oxy_K2_O2 = .000001;
    Oxy_PO2i = Bl_FiO2*Oxy_Pref/100;
    PO2_Oxy_Gas:t = ( Bl_Fgas*(Oxy_PO2i - PO2_Oxy_Gas)*Oxy_K1_O2 +
                    Oxy_Pref * Oxy_DO2 * (PO2_HLM_4-PO2_Oxy_Gas) ) / (Oxy_Vgas);
//Oxygen mix by blender
140    PO2_HLM_4:t =if(HLM_F4>0) HLM_F4*(CtO2_HLM_1-CtO2_HLM_4)/
                ((HLM_V4)*(SolOxy+HbO2Content * cHb *dSHbO2_HLM_4)) +
                Oxy_Pref * Oxy_DO2 *Oxy_K2_O2* ( PO2_Oxy_Gas-PO2_HLM_4 ) /
                (HLM_V4)
                else Oxy_Pref * Oxy_DO2 * ( PO2_Oxy_Gas-PO2_HLM_4 ) / (HLM_V4+Oxy_Vgas);
    when(t = t.min){
        PO2_Oxy_Gas = 608.00    ; // PO2_pa__init
    }
//-----
//          CO2
150 //-----
    private real GAS_alphaCO2 = 24*1.36e-6 M/mmHg; // Solubility constant of CO2 in plasma
                // (assume that is is the same for red blood cells)
    private real P50_CO2 = 250 mmHg;    // P50 of CO2 for Hb
//BLENDER CO2
    real PCO2_Oxy_Gas(t) mmHg;    // Oxygen partial pressure in Oxygenator
    when(t = t.min){
        PCO2_Oxy_Gas = -0.00056388027000000001    ; // PO2_pa__init
    }
//VARIABLES
160    real PCO2_SYS_1(t) ;
    real PCO2_SYS_2(t) ;
    real PCO2_SYS_3(t) ;
    real PCO2_PM_1(t) ;
    real PCO2_HLM_1(t) ;
    real PCO2_HLM_4(t) ;
    real CtCO2_SYS_1(t) ;
    real CtCO2_SYS_2(t) ;
    real CtCO2_SYS_3(t) ;
    real CtCO2_PM_1(t) ;
170    real CtCO2_HLM_1(t) ;
    real CtCO2_HLM_4(t) ;

    real SHbCO2_SYS_1(t) ;
    real SHbCO2_SYS_2(t) ;
    real SHbCO2_SYS_3(t) ;
    real SHbCO2_PM_1(t) ;
    real SHbCO2_HLM_1(t) ;
    real SHbCO2_HLM_4(t) ;
    real dSHbCO2_SYS_1(t) ;
```

```

180  real dSHbCO2_SYS_2(t) ;
      real dSHbCO2_SYS_3(t) ;
      real dSHbCO2_PM_1(t) ;
      real dSHbCO2_HLM_1(t) ;
      real dSHbCO2_HLM_4(t) ;

      SHbCO2_SYS_1= (PCO2_SYS_1/P50_CO2)/((PCO2_SYS_1/P50_CO2) + 1);
      SHbCO2_SYS_2= (PCO2_SYS_2/P50_CO2)/((PCO2_SYS_2/P50_CO2) + 1);
      SHbCO2_SYS_3= (PCO2_SYS_3/P50_CO2)/((PCO2_SYS_3/P50_CO2) + 1);
      SHbCO2_PM_1= (PCO2_PM_1/P50_CO2)/((PCO2_PM_1/P50_CO2) + 1);
190  SHbCO2_HLM_1= (PCO2_HLM_1/P50_CO2)/((PCO2_HLM_1/P50_CO2) + 1);
      SHbCO2_HLM_4= (PCO2_HLM_4/P50_CO2)/((PCO2_HLM_4/P50_CO2) + 1);

      dSHbCO2_SYS_1 = (PCO2_SYS_1^-1) * (PCO2_SYS_1/P50_CO2) / (((PCO2_SYS_1 / P50_CO2)) + 1) ^ 2;
      dSHbCO2_SYS_2 = (PCO2_SYS_2^-1) * (PCO2_SYS_2/P50_CO2) / (((PCO2_SYS_2 / P50_CO2)) + 1) ^ 2;
      dSHbCO2_SYS_3 = (PCO2_SYS_3^-1) * (PCO2_SYS_3/P50_CO2) / (((PCO2_SYS_3 / P50_CO2)) + 1) ^ 2;
      dSHbCO2_PM_1 = (PCO2_PM_1^-1) * (PCO2_PM_1 / P50_CO2) / (((PCO2_PM_1 / P50_CO2)) + 1) ^ 2;
      dSHbCO2_HLM_1 = (PCO2_HLM_1^-1) * (PCO2_HLM_1/P50_CO2) / (((PCO2_HLM_1 / P50_CO2)) + 1) ^ 2;
      dSHbCO2_HLM_4 = (PCO2_HLM_4^-1) * (PCO2_HLM_4/P50_CO2) / (((PCO2_HLM_4 / P50_CO2)) + 1) ^ 2;

200  CtCO2_SYS_1=GAS_alphaCO2*PCO2_SYS_1+CHb*SHbCO2_SYS_1*Hcrit;
      CtCO2_SYS_2=GAS_alphaCO2*PCO2_SYS_2+CHb*SHbCO2_SYS_2*Hcrit;
      CtCO2_SYS_3=GAS_alphaCO2*PCO2_SYS_3+CHb*SHbCO2_SYS_3*Hcrit;
      CtCO2_PM_1=GAS_alphaCO2*PCO2_PM_1+CHb*SHbCO2_PM_1*Hcrit;
      CtCO2_HLM_1=GAS_alphaCO2*PCO2_HLM_1+CHb*SHbCO2_HLM_1*Hcrit;
      CtCO2_HLM_4=GAS_alphaCO2*PCO2_HLM_4+CHb*SHbCO2_HLM_4*Hcrit;

      PCO2_SYS_1:t = (COutput*(PCO2_PM_1-PCO2_SYS_1)+HLM_F4*(PCO2_HLM_4-PCO2_SYS_1))/
                    (SYS_V1+SUP_V1);
      PCO2_SYS_3:t = ((COutput+HLM_F4)*(PCO2_SYS_2-PCO2_SYS_3))/(SYS_V3+SUP_V3);
210  PCO2_PM_1:t = ((COutput)*(PCO2_SYS_3-PCO2_PM_1))/((PM_V1));
      PCO2_HLM_1:t = ((HLM_F4)*(PCO2_SYS_3-PCO2_HLM_1))/((HLM_V1));

      when(t = t.min){
        PCO2_SYS_1 = 69.29050123999999755142 ;
        PCO2_SYS_2 = 69.29023999999999716692 ;
        PCO2_SYS_3 = 69.28749279000000171891 ;
        PCO2_PM_1 = 69.28715658000000132688 ;
        PCO2_HLM_1 = 0 ;
        PCO2_HLM_4 = 0 ;
220  }
      //CO2 release
      real CtCO2_isf(t) M; // Needed for GO2isf - added by mneal
      real PCO2_isf(t) mmHg; // O2 partial pressure in interstitial fluid
      real V_CO2(t) mole/min; // CO2 generation
      private real RQ = 0.8 dimensionless; // Respiratory quotient
      //Saturations
      CtCO2_isf = GAS_alphaCO2*PCO2_isf;
      V_CO2 = RQ*V_O2;
      PCO2_isf:t = PS*PARAM_tweight*( PCO2_SYS_2-PCO2_isf)/Visf + V_CO2/(GAS_alphaCO2*Visf);
230  //---->>
      PCO2_SYS_2:t = (COutput+HLM_F4)*(PCO2_SYS_1-PCO2_SYS_2)/(SYS_V2+SUP_V2) +
                    (PS*PARAM_tweight*GAS_alphaO2*(PCO2_isf-PCO2_SYS_2)*GAS_CO2_PROD)/
                    ((SYS_V2+SUP_V2)*(GAS_alphaCO2+Hcrit*CHb*dSHbCO2_SYS_2));

      when(t=t.min) {
        PCO2_isf = 69.80453106000000218501 ;
      }

      //CO2 remove
240  private real Oxy_PCO2i = 0 mmHg; // Oxygen input partial pressure in Oxygenator
      PCO2_Oxy_Gas:t = ( B1_Fgas * (Oxy_PCO2i - PCO2_Oxy_Gas) +
                      Oxy_Pref * Oxy_DCO2 * (PCO2_HLM_4 - PCO2_Oxy_Gas) ) /
                      (HLM_V4+Oxy_Vgas);
      //---->>

```

C. Mathematical Model (MML)

```
PCO2_HLM_4:t = if (HLM_F4>0) HLM_F4*(PCO2_HLM_1-PCO2_HLM_4)/(HLM_V4) +
    Oxy_Pref * Oxy_DCO2 * (PCO2_Oxy_Gas-PCO2_HLM_4) /
    (HLM_V4+Oxy_Vgas)
    else Oxy_Pref+Oxy_DCO2*(PCO2_Oxy_Gas-PCO2_HLM_4)/(HLM_V4+Oxy_Vgas);

250 // Derived Parameters:

//VO2= Oxygen Consumption
//DO2= Oxygen Delivery
//real VO2(t) mL/kg/min;
//real DO2(t) mL/min/m^2;
//VO2=(COutput+HLM_F4)*(CtO2_SYS_1-CtO2_SYS_3)/PARAM_tweight;
//DO2=(COutput+HLM_F4)*CtO2_SYS_1/PARAM_bsa;
real VO2(t) mL/min;
real DO2(t) mL/min;
260 VO2=(CO+HLM_F4)*(CtO2_SYS_1-CtO2_SYS_3);
DO2=(CO+HLM_F4)*CtO2_SYS_1;

//Effects Temp

extern real PARAM_Temp;
real Viscosity(t);
real kVisc = 1;
Viscosity=0.0394*PARAM_Temp-1.22+0.127*((19+hct_SYS_2*100*(1 dL/g))/PARAM_Temp)^3.88;
//SYS_R1=SYS_R1_0*Viscosity*kVisc;
//SYS_C1=SYS_C1_0;
270 SYS_R1=SYS_R1_0*(1+v_vct_SYS_1*MEDS_Kr_vct_sa-v_vdt_SYS_1*MEDS_Kr_vdt_sa+Viscosity*kVisc);
SYS_C1=SYS_C1_0*(1-v_vct_SYS_1*MEDS_Kc_vct_sa+v_vdt_SYS_1*MEDS_Kc_vdt_sa);

}
```

7. Medication

```
i>>://-----
/*
      X      X  XXXXXXXX  XXXXXXXX  XXXXXX
      XX     XX  X        X        X  X
      X X    X X  X        X        X  X
      X X X  X  XXXXXXXX  X        X  XXXXXX
      X X   X  X        X        X        X
      X     X  X        X        X        X
      X     X  XXXXXXXX  XXXXXXXX  XXXXXX
10  */
//-----
//-----
// Variables
//-----
// Volumes
real ct_SYS_1(t);
real ct_SYS_2(t);
real ct_SYS_3(t);
real ct_PM_1(t);
20  real ct_HLM_1(t);
real ct_HLM_4(t);

real hct_SYS_1(t) g/dL;
real hct_SYS_2(t) g/dL;
real hct_SYS_3(t) g/dL;
real hct_PM_1(t) g/dL;
real hct_HLM_1(t) g/dL;
real hct_HLM_4(t) g/dL;
```

```

30  when(t = t.min){
      ct_SYS_1 = 284 ;
      ct_SYS_2 = 123 ;
      ct_SYS_3 = 1198 ;
      ct_PM_1 = 170 ;
      ct_HLM_1 =231 ;
      ct_HLM_4 = 128 ;
    }
  //Hcrt Change
      hct_SYS_1 = ct_SYS_1/(SYS_V1+SUP_V1) ;
40  hct_SYS_2 = ct_SYS_2/(SYS_V2+SUP_V2) ;
      hct_SYS_3 = ct_SYS_3/(SYS_V3+SUP_V3) ;
      hct_PM_1 = ct_PM_1/PM_V1 ;
      hct_HLM_1 = ct_HLM_1/HLM_V1 ;
      hct_HLM_4 = ct_HLM_4/HLM_V4 ;
      ct_SYS_1:t = ct_PM_1*CO/PM_V1 + ct_HLM_4*EFR/HLM_V4 - ct_SYS_1*(CO+EFR)/(SYS_V1+SUP_V1);
      ct_SYS_2:t = ct_SYS_1*(CO+EFR)/(SYS_V1+SUP_V1)-ct_SYS_2*(CO+EFR)/(SYS_V2+SUP_V2);
      ct_SYS_3:t = ct_SYS_2*(CO+EFR)/(SYS_V2+SUP_V2)-
                  ct_SYS_3*(CO+EFR)/(SYS_V3+SUP_V3) -
                  ct_SYS_3*(ILL_FVOL_LOSS)/(SYS_V3+SUP_V3);
50  ct_PM_1:t = ct_SYS_3*CO/(SYS_V3+SUP_V3) - ct_PM_1*CO/PM_V1;
      ct_HLM_1:t = ct_SYS_3*EFR/(SYS_V3+SUP_V3) - ct_HLM_1*EFR/HLM_V1 ;
      ct_HLM_4:t = ct_HLM_1*EFR/HLM_V1 - ct_HLM_4*EFR/HLM_V4 ;

      real ct_TOTAL(t);

      ct_TOTAL=ct_SYS_1+ct_SYS_2+ct_SYS_3+ct_PM_1+ct_HLM_1+ct_HLM_4;
      ctPlasma_SYS_3=SYS_V3-ct_SYS_3*(1 dL/g);

  //-----
60  // VASOCONSTRICTOR
  //-----
      real MEDS_kCT = 0.13; // Ln2 / t1/2
  //-----
  // Variables
  //-----
  // Volumes
      extern real MEDS_v_vct_IN(t) ml/s;
      real v_vct_SYS_1(t) ml;
      real v_vct_SYS_2(t) ml;
70  real v_vct_SYS_3(t) ml;
      real v_vct_PM_1(t) ml;
      real v_vct_HLM_1(t) ml;
      real v_vct_HLM_4(t) ml;
      real c_vct_SYS_1(t) ;
      real c_vct_SYS_2(t) ;
      real c_vct_SYS_3(t) ;
      real c_vct_PM_1(t) ;
      real c_vct_HLM_1(t) ;
      real c_vct_HLM_4(t) ;
80  when(t = t.min){
      v_vct_SYS_1 = 0.45000000000000000000 ;
      v_vct_SYS_2 = 0.45000000000000000000 ;
      v_vct_SYS_3 = 0.45000000000000000000 ;
      v_vct_PM_1 = 0.45000000000000000000 ;
      v_vct_HLM_1 = 0.00 ;
      v_vct_HLM_4 = 0.00 ;
    }
      c_vct_SYS_1 =v_vct_SYS_1 / (SYS_V1+SUP_V1);
      c_vct_SYS_2 =v_vct_SYS_2 / (SYS_V2+SUP_V2);
90  c_vct_SYS_3 =v_vct_SYS_3 / (SYS_V3+SUP_V3);
      c_vct_PM_1 =v_vct_PM_1 / PM_V1;
      c_vct_HLM_1 =v_vct_HLM_1 / HLM_V1;
      c_vct_HLM_4 =v_vct_HLM_4 / HLM_V4;
  //Volume Change

```

C. Mathematical Model (MML)

```
private real Vct_dif = 0.25;
v_vct_SYS_1:t = (c_vct_PM_1*CO + c_vct_HLM_4*EFR - c_vct_SYS_1*(CO+EFR))*Vct_dif;
v_vct_SYS_2:t = (c_vct_SYS_1*(CO+EFR)-c_vct_SYS_2*(CO+EFR))*Vct_dif;
v_vct_SYS_3:t = (c_vct_SYS_2*(CO+EFR)-c_vct_SYS_3*(CO+EFR))*Vct_dif-MEDS_kCT*c_vct_SYS_2;
v_vct_PM_1:t = (c_vct_SYS_3*CO - c_vct_PM_1*CO)*Vct_dif;
100 v_vct_HLM_1:t = (c_vct_SYS_3*EFR - c_vct_HLM_1*EFR)*Vct_dif + MEDS_v_vct_IN;
v_vct_HLM_4:t = (c_vct_HLM_1*EFR - c_vct_HLM_4*EFR)*Vct_dif ;
real v_vct_TOTAL(t);
v_vct_TOTAL=v_vct_SYS_1+v_vct_SYS_2+v_vct_SYS_3+v_vct_PM_1+v_vct_HLM_1+v_vct_HLM_4;
//-----
// VASODILATOR
//-----
real MEDS_kDT = 0.13; // Ln2 / t1/2
//-----
// Variables
110 //-----
// Volumes
extern real MEDS_v_vdt_IN(t) ml/s;
real v_vdt_SYS_1(t) ml;
real v_vdt_SYS_2(t) ml;
real v_vdt_SYS_3(t) ml;
real v_vdt_PM_1(t) ml;
real v_vdt_HLM_1(t) ml;
real v_vdt_HLM_4(t) ml;
120 real c_vdt_SYS_1(t) ;
real c_vdt_SYS_2(t) ;
real c_vdt_SYS_3(t) ;
real c_vdt_PM_1(t) ;
real c_vdt_HLM_1(t) ;
real c_vdt_HLM_4(t) ;
when(t = t.min){
v_vdt_SYS_1 = 0.00 ;
v_vdt_SYS_2 = 0.00 ;
v_vdt_SYS_3 = 0.00 ;
v_vdt_PM_1 = 0.00 ;
130 v_vdt_HLM_1 = 0.00 ;
v_vdt_HLM_4 = 0.00 ;
}
c_vdt_SYS_1 =v_vdt_SYS_1 / (SYS_V1+SUP_V1);
c_vdt_SYS_2 =v_vdt_SYS_2 / (SYS_V2+SUP_V2);
c_vdt_SYS_3 =v_vdt_SYS_3 / (SYS_V3+SUP_V3);
c_vdt_PM_1 =v_vdt_PM_1 / PM_V1;
c_vdt_HLM_1 =v_vdt_HLM_1 / HLM_V1;
c_vdt_HLM_4 =v_vdt_HLM_4 / HLM_V4;
//Volume Change
140 private real Vdt_dif = 0.286;
v_vdt_SYS_1:t = (c_vdt_PM_1*CO + c_vdt_HLM_4*EFR - c_vdt_SYS_1*(CO+EFR))*Vdt_dif;
v_vdt_SYS_2:t = (c_vdt_SYS_1*(CO+EFR)-c_vdt_SYS_2*(CO+EFR))*Vdt_dif;
v_vdt_SYS_3:t = (c_vdt_SYS_2*(CO+EFR)-c_vdt_SYS_3*(CO+EFR))*Vdt_dif-MEDS_kDT*c_vdt_SYS_2;
v_vdt_PM_1:t = (c_vdt_SYS_3*CO - c_vdt_PM_1*CO)*Vdt_dif;
v_vdt_HLM_1:t = (c_vdt_SYS_3*EFR - c_vdt_HLM_1*EFR)*Vdt_dif + MEDS_v_vdt_IN;
v_vdt_HLM_4:t = (c_vdt_HLM_1*EFR - c_vdt_HLM_4*EFR)*Vdt_dif ;
real v_vdt_TOTAL(t);
v_vdt_TOTAL=v_vdt_SYS_1+v_vdt_SYS_2+v_vdt_SYS_3+v_vdt_PM_1+v_vdt_HLM_1+v_vdt_HLM_4;
//-----
150 // Effects
//-----
real MEDS_Kr_vct_sa = 1.00 ;
real MEDS_Kc_vct_sa = 1.00 ;
real MEDS_Kr_vdt_sa = 1.00 ;
real MEDS_Kc_vdt_sa = 1.00 ;
//SYS_R1=SYS_R1_0*(1+v_vct_SYS_1*MEDS_Kr_vct_sa-v_vdt_SYS_1*MEDS_Kr_vdt_sa);
//SYS_C1=SYS_C1_0*(1+v_vct_SYS_1*MEDS_Kc_vct_sa-v_vdt_SYS_1*MEDS_Kc_vdt_sa);
//SYS_R1=SYS_R1_0;
//SYS_C1=SYS_C1_0;
```

D

Appendix

ECSS control

This section contains the configuration files that were used with the software framework for the automation of the ECSS. For evaluation purposes the application was connected with the simulation environment through a network communication as described in III.3.1.d.

1. App.xml

```
<Application Name="ECSS_Control">
  <MDC>
    <LoadConfig Path="Config/MDC" W="565" H="350" FullScreen="false" RefreshInterval="100" />
    <Objects File="MDC/Parameters.xml" />
    <!--Net objects are used with simulation -->
    <Objects File="MDC/NetObjects.xml" />
    <Objects File="MDC/PumpObjects.xml" />
    <Objects File="MDC/ScopesObjects.xml" />
    <Objects File="MDC/MedsObjects.xml" />
    <Objects File="MDC/BlenderObjects.xml" />
    <Objects File="MDC/SVRObjects.xml" />
    <Objects File="MDC/SupervisorObjects.xml" />
    <Lib File="AutoMedic_OBJ" />
    <Lib File="AutoMedic_NFC" />
    <Lib File="AutoMedic_QT" />
  </MDC>
</Application>
```

10

```
</MDC>
</Application>
```

2. Parameters.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<MDC Name="Parameters">
  <GroupObject Name="Patient" Icon="Icons/patientIcon.ico">
    <Parameters Name="MainParameters" Icon="Icons/patientIcon.ico" PW="800" PH="600">
      <InParameter Id="0" Name="MAP" Units="mmHg"
        CurrentValue="cMAP" CriticalMin="40" Min="50" Normal="60" Max="90" CriticalMax="110"
        Weight="1" RefModel="true" EnablePort="PumpControlEnable">
        <RefModel IntervalUp="0.01" IntervalDown="0.01" DelayUp="0" DelayDown="0" />
      </InParameter>
      <InParameter Id="1" Name="EFR" Units="L/min"
        CurrentValue="cEFR" CriticalMin="-1" Min="2" Normal="4.5" Max="5" CriticalMax="6"
        Weight="1" RefModel="true" EnablePort="PumpControlEnable" >
        <RefModel IntervalUp="0.01" IntervalDown="0.01" DelayUp="0" DelayDown="0" />
      </InParameter>
      <InParameter Id="2" Name="SVR" Units=""
        CurrentValue="fSVR" CriticalMin="5" Min="10" Normal="2500" Max="2500" CriticalMax="3000"
        Weight="1" RefModel="true" EnablePort="MedsControlEnable" >
        <RefModel IntervalUp="0.005" IntervalDown="0.0005" DelayUp="0" DelayDown="0" />
      </InParameter>
      <OutParameter Id="0" Name="PumpOut" Units="fpm" Icon="Icons/Pump.png" Increase="true"
        CriticalMin="999" Min="1000" Normal="1275.44" Max="3900" CriticalMax="3901" Weight="1"
        cName="cPumpOut" dName="dPumpOut" />
    </Parameters>
    <Parameters Name="GasParameters" Icon="Icons/patientIcon.ico" PW="800" PH="600" >
      <InParameter Id="0" Name="VO2" Units=""
        CurrentValue="cVO2" CriticalMin="100" Min="200" Normal="500" Max="1200" CriticalMax="2400"
        Weight="0" RefModel="true" EnablePort="GasControlEnable" >
        <RefModel IntervalUp="0.01" IntervalDown="0.01" DelayUp="0" DelayDown="0" />
      </InParameter>
      <InParameter Id="1" Name="DO2" Units=""
        CurrentValue="cDO2" CriticalMin="200" Min="400" Normal="750" Max="1200" CriticalMax="2400"
        Weight="0" RefModel="true" EnablePort="GasControlEnable" >
        <RefModel IntervalUp="0.01" IntervalDown="0.01" DelayUp="0" DelayDown="0" />
      </InParameter>
      <InParameter Id="2" Name="SpO2a" Units="%"
        CurrentValue="cSpO2a" CriticalMin="70" Min="85" Normal="100" Max="100" CriticalMax="101"
        Weight="1" RefModel="true" EnablePort="GasControlEnable" >
        <RefModel IntervalUp="0.1" IntervalDown="0.005" DelayUp="0" DelayDown="0" />
    </Parameters>
```



```

40 </InParameter>
<InParameter Id="3" Name="SpO2v" Units="%"
  CurrentValue="cSpO2v" CriticalMin="30" Min="35" Normal="75" Max="90" CriticalMax="100"
  Weight="1" RefModel="true" EnablePort="GasControlEnable" >
  <RefModel IntervalUp="0.1" IntervalDown="0.005" DelayUp="0" DelayDown="0" />
</InParameter>
<InParameter Id="4" Name="PO2a" Units="mmHg"
  CurrentValue="cPO2a" CriticalMin="70" Min="90" Normal="100" Max="400" CriticalMax="500"
  Weight="0.5" RefModel="true" EnablePort="GasControlEnable" >
  <RefModel IntervalUp="0.5" IntervalDown="0.005" DelayUp="0" DelayDown="0" />
</InParameter>
<InParameter Id="5" Name="PO2v" Units="mmHg"
  CurrentValue="cPO2v" CriticalMin="25" Min="30" Normal="50" Max="60" CriticalMax="65"
  Weight="0.5" CurrentValue="cPO2v" RefModel="true" >
  <RefModel IntervalUp="0.005" IntervalDown="0.05" DelayUp="0" DelayDown="0" />
</InParameter>
<InParameter Id="6" Name="PCO2a" Units="mmHg"
  CurrentValue="cPCO2a" CriticalMin="20" Min="30" Normal="40" Max="50" CriticalMax="60"
  Weight="0.5" RefModel="true" EnablePort="GasControlEnable" >
  <RefModel IntervalUp="0.1" IntervalDown="0.005" DelayUp="0" DelayDown="0" />
</InParameter>
<InParameter Id="7" Name="PCO2v" Module="2" Units="mmHg"
  CurrentValue="cPCO2v" CriticalMin="25" Min="30" Normal="42" Max="50" CriticalMax="60"
  Weight="1" RefModel="true" >
  <RefModel IntervalUp="0.1" IntervalDown="0.005" DelayUp="0" DelayDown="0" />
</InParameter>
<InParameter Id="8" Name="Hcrit" Units="
  CurrentValue="cHcrit" CriticalMin="5" Min="10" Normal="35" Max="50" CriticalMax="1900"
  Weight="1" RefModel="true" >
  <RefModel IntervalUp="0.01" IntervalDown="0.01" DelayUp="0" DelayDown="0" />
</InParameter>
<InParameter Id="9" Name="PumpGasCorrection" Units="
  CriticalMin="0" Min="1" Normal="1" Max="1" CriticalMax="2" Weight="1" RefModel="false" />
<OutParameter Id="0" Name="FiO2Out" Units="%" Icon="Icons/Blender.png" Increase="true"
  CriticalMin="19" Min="20" Normal="81.9316" Max="100" CriticalMax="101" Weight="1" cName="dFiO2Out" dName="dFiO2Out" />
<OutParameter Id="1" Name="GasFlowOut" Units="L/min" Icon="Icons/Blender.png" Increase="true"
  CriticalMin="1" Min="2" Normal="10" Max="10" CriticalMax="11" Weight="1" cName="cGasFlowOut" dName="dGasFlowOut" />
<OutParameter Id="2" Name="VolOut" Units="ml/min" Increase="true"
  CriticalMin="-1" Min="0" Normal="0" Max="200" CriticalMax="20" Weight="1" cName="cVolOut" dName="dVolOut" />
<OutParameter Id="3" Name="Meds" Units="ug/d" Increase="true"
  CriticalMin="-10" Min="-9" Normal="9" Max="9" CriticalMax="10" Weight="1" cName="cMeds" dName="dMeds" Icon="UNDEF" />
80 </Parameters>
</GroupObject>
</MDC>

```

3. PumpObjects.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<MDC>
  <Calculate Trigger="NetInADC" >
    <Object Name="WeightSum1" />
    <Object Name="PumpAdaptiveFcb_DIS" />
    <Object Name="PumpOut" />
    <Object Name="NetOutPump" />
  </Calculate>
  <GroupObject Name="Pump Control" Icon="Icons/LifebridgeBW.png" PW="988" PH="456" >
    <MSE Name="Pump_MSE" Precision="4" >
      <Var Name="nt_eMAP" WeightName="wMAP" />
      <Var Name="nt_eEFR" WeightName="wEFR" />
    </MSE>
    <Filter Name="FilterPSA" Input="cPSA" Output="cMAP" NormFreq="0.1" CenterFreq="0.1" MultFactor="1" />
    <Filter Name="FilterEF" Input="cEF" Output="cEFR" NormFreq="0.1" CenterFreq="0.1" MultFactor="10" />
    <FuzzyControl Name="cMAPb" Inputs="3" Outputs="1" Type="Mamdani" EnablePort="PumpControlEnable" >
      <Input Id="0" Name="r_eMAP" NumSets="5" Gain="0.025" CreateSets="true"
        Setstype="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
      <Input Id="1" Name="r_deMAP" NumSets="5" Gain="0.1" CreateSets="true"
        Setstype="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
      <Input Id="2" Name="t_eMAP" NumSets="3" Gain="0.1" CreateSets="true"
        Setstype="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
      <Output Id="0" Name="dPump0" Gain="0.03" MakeRules="PI" Inverse="true" Distrib="Regular" />
    </FuzzyControl>
    <FuzzyControl Name="c2EFRb" Inputs="3" Outputs="1" Type="Mamdani" EnablePort="PumpControlEnable" >
      <Input Id="0" Name="r_eEFR" NumSets="5" Gain="0.25" CreateSets="true"
        Setstype="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
      <Input Id="1" Name="r_deEFR" NumSets="5" Gain="0.1" CreateSets="true"
        Setstype="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
      <Input Id="2" Name="t_eEFR" NumSets="3" Gain="0.1" CreateSets="true"
        Setstype="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
      <Output Id="0" Name="dPump1" Gain="0.03" MakeRules="PI" Inverse="true" Distrib="Regular" />
    </FuzzyControl>
    <FuzzyControl Name="c302" Inputs="2" Outputs="1" Type="Mamdani" EnablePort="PumpControlEnable" >
      <Input Id="0" Name="vO2" NumSets="5" Gain="0.25" CreateSets="true"
        Setstype="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
      <Input Id="1" Name="DO2" NumSets="5" Gain="0.1" CreateSets="true"
        Setstype="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
      <Output Id="0" Name="dPump2" Gain="0.03" MakeRules="PI" Inverse="true" Distrib="Regular"
        NumSets="-1" CreateSets="true" Overlap="100" OnlyPositive="false" />
    </FuzzyControl>
    <FuzzyControl Name="c4PumpGasCorrection" Inputs="5" Outputs="1" Type="Mamdani" EnablePort="PumpControlEnable" >
      <Input Id="0" Name="FIO2Out" NumSets="3" Gain="0.01" CreateSets="true"

```

```

SetsType="TRI" Distrib="Regular" Overlap="80" NumSEts="3" Gain="0.1" CreateSets="true" />
<Input Id="1" Name="GasFlowOut" NumSEts="3" Gain="0.1" CreateSets="true" />
SetsType="TRI" Distrib="Regular" Overlap="80" NumSEts="3" Gain="0.1" CreateSets="true" />
<Input Id="2" Name="nr_ePCO2y" NumSEts="3" Gain="1" CreateSets="true" />
SetsType="TRI" Distrib="Regular" Overlap="80" NumSEts="3" Gain="1" CreateSets="true" />
<Input Id="3" Name="nr_eSpO2y" NumSEts="3" Gain="1" CreateSets="true" />
SetsType="TRI" Distrib="Regular" Overlap="80" NumSEts="3" Gain="1" CreateSets="true" />
<Input Id="4" Name="nr_ePO2a" NumSEts="3" Gain="1" CreateSets="true" />
SetsType="TRI" Distrib="Regular" Overlap="80" NumSEts="3" Gain="1" CreateSets="true" />
<Output Id="0" Name="dPump3" Gain="0.03" Inverse="true" Distrib="Regular" />
</FuzzyControl>
<WeightSum Name="WeightSum1" Output="dPumpOut" >
<Input Name="dPump0" WeightName="wMAP" WeightPos="1" WeightNeg="1" />
<Input Name="dPump1" WeightName="wEFR" WeightPos="1" WeightNeg="1" />
<Input Name="dPump2" WeightName="wVO2" WeightPos="1" WeightNeg="1" />
<Input Name="dPump3" WeightName="wPumpGasCorrection" WeightPos="1" WeightNeg="1" />
</WeightSum>
<WeightSum Name="WeightSum2" Output="ePump" X="516" Y="214" W="77" H="71" DrawInputs="false" >
<Input Name="nr_eMAP" WeightName="wMAP" WeightPos="1" WeightNeg="1" />
<Input Name="nr_eEFR" WeightName="wEFR" WeightPos="1" WeightNeg="1" />
<Input Name="nr_eDO2" WeightName="wDO2" WeightPos="1" WeightNeg="1" />
<Input Name="nr_eVO2" WeightName="wVO2" WeightPos="1" WeightNeg="1" />
</WeightSum>
<Derivative Name="Der1" Input="wePump" Output="dwePump" />
<AdaptiveFuzzyControl Name="PumpAdaptiveFcb_DIS" Inputs="8" Outputs="1" Type="Mamdani"
Learning="true" Adapting="true" Correcting="false" EnablePort="PumpControlEnable" >
<MFOS Threshold="0.3" Error="wePump" DErrror="dwePump" Ref="dPumpOut"
AdaptingGain="10" Id="0" InverseFIM="true" GainError="10" GainError="1" Delay="1" />
<Input Id="0" Name="r_eMAP" NumSEts="9" Gain="0.025" CreateSets="true"
SetsType="TRI" Distrib="Focus" Overlap="95" OnlyPositive="false" />
<Input Id="1" Name="t_eMAP" NumSEts="3" Gain="0.02" CreateSets="true"
SetsType="TRI" Distrib="Focus" Overlap="95" OnlyPositive="false" />
<Input Id="2" Name="r_deMAP" NumSEts="7" Gain="1" CreateSets="true"
SetsType="TRI" Distrib="Focus" Overlap="95" OnlyPositive="false" />
<Input Id="3" Name="r_eEFR" NumSEts="9" Gain="0.25" CreateSets="true"
SetsType="TRI" Distrib="Focus" Overlap="95" OnlyPositive="false" />
<Input Id="4" Name="t_eEFR" NumSEts="3" Gain="0.25" CreateSets="true"
SetsType="TRI" Distrib="Focus" Overlap="95" OnlyPositive="false" />
<Input Id="5" Name="r_deEFR" NumSEts="7" Gain="1" CreateSets="true"
SetsType="TRI" Distrib="Focus" Overlap="95" OnlyPositive="false" />
<Input Id="6" Name="VO2" NumSEts="5" Gain="0.25" CreateSets="true"
SetsType="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
<Input Id="7" Name="DO2" NumSEts="5" Gain="0.1" CreateSets="true"
SetsType="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
<Output Id="0" Name="dPumpOut_DIS" Gain="0.003" Distrib="Focus"

```

```

    NumSets="31" CreateSets="true" Overlap="100" MakeRules=" " />
  </AdaptiveFuzzyControl>
</GroupObject>
</MDC>

```

90

4. BlenderObjects.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<MDC Name="">
  <GroupObject Name="Blender Control" PW="1024" PH="575" >
    <MSE Name="Blender_MSE" Precision="4" >
      <Var Name="nt_eVO2" WeightName="wVO2" />
      <Var Name="nt_eDO2" WeightName="wDO2" />
      <Var Name="nt_eSpO2a" WeightName="wSpO2a" />
      <Var Name="nt_ePO2a" WeightName="wPO2a" />
      <Var Name="nt_ePO2v" WeightName="wPO2v" />
      <Var Name="nt_ePCO2a" WeightName="wPCO2a" />
      <Var Name="nt_ePCO2v" WeightName="wPCO2v" />
    </MSE>
    <FuzzyControl Name="c1FiO2v_c" Inputs="3" Outputs="1" Type="Mamdani" >
      <Input Id="0" Name="r_desPO2v" NumSets="5" Gain="0.1" CreateSets="true"
        SetsType="TRI" Distrib="Regular" Overlap="100" OnlyPositive="false" />
      <Input Id="1" Name="r_eSpO2v" NumSets="5" Gain="0.1" CreateSets="true"
        SetsType="TRI" Distrib="Regular" Overlap="100" OnlyPositive="false" />
      <Input Id="2" Name="t_eSpO2v" NumSets="3" Gain="0.1" CreateSets="true"
        SetsType="TRI" Distrib="Regular" Overlap="100" OnlyPositive="false" />
      <Output Id="0" Name="dFiO2_0" Gain="5" MakeRules="PI" Inverse="true" Distrib="Regular" />
    </FuzzyControl>
    <FuzzyControl Name="c2FiO2a" Inputs="2" Outputs="1" Type="Mamdani" >
      <Input Id="0" Name="r_ePO2a" NumSets="5" Gain="0.01" CreateSets="true"
        SetsType="TRI" Distrib="Regular" Overlap="100" OnlyPositive="false" />
      <Input Id="1" Name="r_dePO2a" NumSets="5" Gain="0.02" CreateSets="true"
        SetsType="TRI" Distrib="Regular" Overlap="100" OnlyPositive="false" />
      <Output Id="0" Name="dFiO2_1" Gain="5" MakeRules="PI" Inverse="true" Distrib="Regular" />
    </FuzzyControl>
    <FuzzyControl Name="c3FiO2" Inputs="2" Outputs="1" Type="Mamdani" >
      <Input Id="0" Name="r_ePO2a" NumSets="5" Gain="0.01" CreateSets="true"
        SetsType="TRI" Distrib="Regular" Overlap="100" OnlyPositive="false" />
      <Input Id="1" Name="r_ePO2v" NumSets="5" Gain="0.02" CreateSets="true"
        SetsType="TRI" Distrib="Regular" Overlap="100" OnlyPositive="false" />
      <Output Id="0" Name="dFiO2_2" Gain="5" MakeRules="PI" Inverse="true" Distrib="Regular" />
    </FuzzyControl>
  </GroupObject>
  <!--This controller checks if the PO2a is too high and Po2v

```

10

20

30

```

is above its
target, then it will decrease FiO2 to avoid
hyperoxiy-->

</Comment>
</FuzzyControl>
<FuzzyControl Name="c1GasFlowb" Inputs="3" Outputs="1" Type="Mamdani" >
  <Input Id="0" Name="r_defPCO2v" NumSets="5" Gain="1" CreateSets="true"
    SetsType="TRI" Distrib="Regular" Overlap="80" OnlyPositive="false" />
  <Input Id="1" Name="r_ePCO2v" NumSets="5" Gain="1" CreateSets="true"
    SetsType="TRI" Distrib="Regular" Overlap="80" OnlyPositive="false" />
  <Input Id="2" Name="t_ePCO2v" NumSets="3" Gain="1" CreateSets="true"
    SetsType="TRI" Distrib="Regular" Overlap="80" OnlyPositive="false" />
  <Output Id="0" Name="dGasFlow_0" Gain="5" MakeRules="PI" Inverse="false" Distrib="Regular" />
</FuzzyControl>
<FuzzyControl Name="c2GasFlow" Inputs="3" Outputs="1" Type="Mamdani" >
  <Input Id="0" Name="cFiO2Out" NumSets="3" Gain="0.01" CreateSets="true"
    SetsType="TRI" Distrib="Regular" Overlap="80" OnlyPositive="true" />
  <Input Id="1" Name="r_eSpO2v" NumSets="5" Gain="0.1" CreateSets="true"
    SetsType="TRI" Distrib="Regular" Overlap="80" OnlyPositive="false" />
  <Input Id="2" Name="r_eSpO2a" NumSets="5" Gain="0.1" CreateSets="true"
    SetsType="TRI" Distrib="Regular" Overlap="80" OnlyPositive="false" />
  <Output Id="0" Name="dGasFlow_1" Gain="5" Inverse="false" Distrib="Regular" />
</Comment>
<Rule Id="0" Text="When FiO2 is already 100% and SpO2v or SpO2a is low then increase gas flow" />
</Comment>
</FuzzyControl>
<WeightSum Name="dFiO2Out" Output="_dFiO2Out" EnablePort="GasControlEnable" >
  <Input Name="dFiO2_0" WeightName="wSpO2v" WeightPos="1" WeightNeg="1" />
  <Input Name="dFiO2_1" WeightName="wPO2a" WeightPos="1" WeightNeg="1" />
  <Input Name="dFiO2_2" Weight="1" WeightName="Undef" WeightPos="1" WeightNeg="1" />
</WeightSum>
<WeightSum Name="dGasFlowOut" Output="_dGasFlowOut" >
  <Input Name="dGasFlow_0" WeightName="wSpO2v" WeightPos="1" WeightNeg="1" />
  <Input Name="dGasFlow_1" WeightName="wSpO2a" WeightPos="1" WeightNeg="1" />
</WeightSum>
<WeightSum Name="WeightSum2" Output="eGasFlow" >
  <Input Name="nr_eSpO2v" WeightName="wSpO2v" WeightPos="0" WeightNeg="1" />
  <Input Name="nr_eSpO2a" WeightName="wSpO2a" WeightPos="0" WeightNeg="1" />
  <Input Name="nr_ePO2a" WeightName="wPO2a" WeightNeg="0" WeightPos="1" />
  <Input Name="nr_ePCO2v" WeightName="wPCO2v" WeightPos="-1" WeightNeg="0" />
</WeightSum>
<WeightSum Name="WeightSum3" Output="eFiO2" >
  <Input Name="nr_eSpO2v" WeightName="wSpO2v" WeightPos="0" WeightNeg="1" />
  <Input Name="nr_ePO2a" WeightName="wPO2a" WeightPos="1" WeightNeg="1" />
</WeightSum>

```

```

<Derivative Name="Der2" Input="weGasFlow" Output="dweGasFlow" />
<Derivative Name="Der3" Input="weFiO2" X="565" Y="161" W="50" H="68" Output="dweFiO2" />
<AdaptiveFuzzyControl Name="GasFlowAdaptiveFC_b" Inputs="11" Outputs="1" Type="Mamdani" Learning="true"
  Adapting="false" Correcting="false" X="704" Y="265" W="117" H="236" EnablePort="GasControlEnable" >
  <MFOS Threshold="0.3" Error="weGasFlow" DError="dweGasFlow" Ref="dGasFlowOut" AdaptingGain="0.1" Id="0"
    InverseFIM="true" GainError="10" GainError="1" Delay="1" />
  <Input Id="0" Name="r_dePCO2v" NumSets="5" Gain="1" CreateSets="true" />
  <Input Id="1" Name="r_ePCO2v" NumSets="5" Gain="1" CreateSets="true" />
  <Input Id="2" Name="t_ePCO2v" NumSets="3" Gain="1" CreateSets="true" />
  <Input Id="3" Name="r_eMAP" NumSets="9" Gain="0.025" CreateSets="true" />
  <Input Id="4" Name="r_eEFR" NumSets="9" Gain="0.25" CreateSets="true" />
  <Input Id="5" Name="r_eSpO2v" NumSets="5" Gain="0.1" CreateSets="true" />
  <Input Id="6" Name="r_deSpO2v" NumSets="5" Gain="0.1" CreateSets="true" />
  <Input Id="7" Name="r_ePO2a" NumSets="5" Gain="0.1" CreateSets="true" />
  <Input Id="8" Name="r_dePO2a" NumSets="5" Gain="0.1" CreateSets="true" />
  <Input Id="9" Name="r_ePO2v" NumSets="5" Gain="0.1" CreateSets="true" />
  <Input Id="10" Name="r_dePO2v" NumSets="5" Gain="0.1" CreateSets="true" />
  <Output Id="0" Name="dGasFlowOut" Gain="1" Distrib="Focus" NumSets="11" />
</AdaptiveFuzzyControl>
<AdaptiveFuzzyControl Name="FiO2AdaptiveFC_b" Inputs="11" Outputs="1" Type="Mamdani" Learning="true"
  Adapting="true" Correcting="false" X="703" Y="12" W="121" H="236" EnablePort="GasControlEnable" >
  <MFOS Threshold="0.3" Delay="10" Error="weFiO2" DError="dweFiO2" Ref="_dFiO2Out" AdaptingGain="0.1"
    InverseFIM="true" GainError="10" GainError="1" />
  <Input Id="0" Name="r_dePCO2v" NumSets="5" Gain="1" CreateSets="true" />
  <Input Id="1" Name="r_ePCO2v" NumSets="5" Gain="1" CreateSets="true" />
  <Input Id="2" Name="t_ePCO2v" NumSets="3" Gain="1" CreateSets="true" />
  <Input Id="3" Name="r_eMAP" NumSets="9" Gain="0.025" CreateSets="true" />
  <Input Id="4" Name="r_eEFR" NumSets="9" Gain="0.25" CreateSets="true" />
  <Input Id="5" Name="r_eSpO2v" NumSets="5" Gain="0.1" CreateSets="true" />

```

90

100

110

120

```

130 SetsType="TRI" Distrib="Regular" Overlap="100" OnlyPositive="false" />
    <Input Id="6" Name="r_deSpO2v" NumSets="5" Gain="0.1" CreateSets="true" />
    SetsType="TRI" Distrib="Regular" Overlap="100" OnlyPositive="false" />
    <Input Id="7" Name="r_ePO2a" NumSets="5" Gain="0.1" CreateSets="true" />
    SetsType="TRI" Distrib="Regular" Overlap="80" OnlyPositive="false" />
    <Input Id="8" Name="r_dePO2a" NumSets="5" Gain="0.1" CreateSets="true" />
    SetsType="TRI" Distrib="Regular" Overlap="80" OnlyPositive="false" />
    <Input Id="9" Name="r_ePO2v" NumSets="5" Gain="0.1" CreateSets="true" />
    SetsType="TRI" Distrib="Regular" Overlap="80" OnlyPositive="false" />
    <Input Id="10" Name="r_dePO2v" NumSets="5" Gain="0.1" CreateSets="true" />
    SetsType="TRI" Distrib="Regular" Overlap="80" OnlyPositive="false" />
    <Output Id="0" Name="dFlO2Out" Gain="1" Distrib="Focus" NumSets="11" />
  </AdaptiveFuzzyControl>
</GroupObject>
140 </MDC>
</MDC>

5. MedsObjects.xml

<?xml version="1.0" encoding="UTF-8" ?>
<MDC Name="">
  <GroupObject Name="Meds Control" PW="1024" PH="575" >
    <FuzzyControl Name="MEDS" Inputs="3" Outputs="1" Type="Mamdani" EnablePort="MedsControlEnable" >
      <Input Id="0" Name="r_eSVR" NumSets="5" Gain="0.25" CreateSets="true"
        SetsType="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
      <Input Id="1" Name="r_deSVR" NumSets="5" Gain="0.1" CreateSets="true"
        SetsType="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
      <Input Id="2" Name="t_eSVR" NumSets="3" Gain="0.1" CreateSets="true"
        SetsType="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
      <Output Id="0" Name="_dMeds" Gain="1" MakeRules="PI" Inverse="true" Distrib="Regular" />
    </FuzzyControl>
    <Saturation Name="SatVct" Output="_vct" Input="Meds" Min="0" Max="1000" />
    <Saturation Name="SatVdt" Output="_vdt" Input="Meds" Min="-1000" Max="0" />
    <Gain Name="GISDN" Output="ISDN" Input="_vct" Gain="1" />
    <Gain Name="GNEP" Output="NEP" Input="_vdt" Gain="-1" />
    <Memory Name="MemMeds" Output="cMeds" Input="Meds" />
    <Constant Name="dVolOut" Value="0" />
    <WeightSum Name="WeightSumMeds" Output="eMeds" >
      <Input Name="nr_eSVR" WeightName="wSVR" WeightPos="1" WeightNeg="1" />
    </WeightSum>
    <Derivative Name="Der1" Input="weMeds" Output="dweMeds" />
    <AdaptiveFuzzyControl Name="MedsAdaptiveFCb" Inputs="8" Outputs="1" Type="Mamdani" Learning="true"
      Adapting="false" Correcting="false" X="621" Y="60" W="120" H="220" EnablePort="MedsControlEnable" >
      <MFOS Threshold="0.3" Error="weMeds" DError="dweMeds" Ref="_dMeds" Id="0" AdaptingGain="1"
10
20

```

```

InverseFIM="true" GainError="10" GainError="1" Delay="1" />
<Input Id="0" Name="r_eSVR" NumSEts="5" Gain="0.25" CreateSEts="true"
  SEtsType="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
<Input Id="1" Name="r_desVR" NumSEts="5" Gain="0.1" CreateSEts="true"
  SEtsType="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
<Input Id="2" Name="t_eSVR" NumSEts="3" Gain="0.1" CreateSEts="true"
  SEtsType="TRI" Distrib="Focus" Overlap="80" OnlyPositive="false" />
<Input Id="3" Name="t_eMAP" NumSEts="3" Gain="0.02" CreateSEts="true"
  SEtsType="TRI" Distrib="Focus" Overlap="95" OnlyPositive="false" />
<Input Id="4" Name="r_deMAP" NumSEts="7" Gain="1" CreateSEts="true"
  SEtsType="TRI" Distrib="Focus" Overlap="95" OnlyPositive="false" />
<Input Id="5" Name="r_eEFR" NumSEts="9" Gain="0.25" CreateSEts="true"
  SEtsType="TRI" Distrib="Focus" Overlap="95" OnlyPositive="false" />
<Input Id="6" Name="t_eEFR" NumSEts="3" Gain="0.25" CreateSEts="true"
  SEtsType="TRI" Distrib="Focus" Overlap="95" OnlyPositive="false" />
<Input Id="7" Name="r_deEFR" NumSEts="7" Gain="1" CreateSEts="true"
  SEtsType="TRI" Distrib="Focus" Overlap="95" OnlyPositive="false" />
<Output Id="0" Name="dMeds" Gain="5" Distrib="Focus" NumSEts="31" />
</AdaptiveFuzzyControl>
</GroupObject>
</MDC>

```

30

40

6. SVRObjects.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<MDC Name="">
  <GroupObject Name="Meds Control" W="1024" PH="575" >
    <GroupObject Name="SVR Calculation" PW="800" PH="546" >
      <Inverse Name="Inv_tEFR" Input="tEFR" Output="1/tEFR" />
      <Inverse Name="Inv_cEFR" Input="cEFR" Output="1/cEFR" />
      <Mult Name="ct_SVR" X="256" Y="82" W="50" H="80" >
        <Input Name="t_MAP" />
        <Input Name="1/tEFR" />
        <Input Value="80" Name="Undef" />
      </Mult>
      <Mult Name="cSVR" X="255" Y="187" W="60" H="80" >
        <Input Name="cMAP" />
        <Input Name="1/cEFR" />
        <Input Value="80" Name="Undef" />
      </Mult>
      <Filter Name="FilterSVR" Input="cSVR" Output="fSVR" NormFreq="0.1"
        CenterFreq="0.1"MultFactor="1" />
    </GroupObject>

```

10


```

20 </GroupObject>
    </MDC>

```

7. NetObjects.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<MDC Name="Network">
  <Client Name="NET" X="22" Y="143" IP="localhost" Port="3000"AutoConnect="true" />
  <!-- <Server Name="Server" Icon="Icons/networkSmall.ico" X="0" Y="430" Port="3000" W="63" H="62" /> -->
  <GroupObject Name="Pump Control" Icon="Icons/Lifebridge.png"PW="988" PH="456" ResetPort="Undef">
    <InputPacket Name="NetInADC" Icon="Icons/Patient.png" Source="NET" Channels="I2" Freq="1" Samples="1" TriggerFreq="1">
      <Chan Id="0" Name="CPSA" Units="mmHg" />
      <Chan Id="1" Name="cCVP" Units="mmHg" />
      <Chan Id="2" Name="cEF" Units="L/min" />
      <Chan Id="3" Name="cCO" Units="L/min" />
      <Chan Id="4" Name="cSVR0" Units="rpm" />
      <Chan Id="5" Name="CHR" Units="rpm" />
      <Chan Id="6" Name="cVO2" Units="rpm" />
      <Chan Id="7" Name="cDO2" Units="rpm" />
      <Chan Id="8" Name="cPumpOut" Units="rpm" />
      <Chan Id="9" Name="cPin" Units="rpm" />
      <Chan Id="10" Name="cPOxy" Units="rpm" />
      <Chan Id="11" Name="cPOut" Units="rpm" />
    </InputPacket>
    <OutputPacket Module="10" Name="NetOutPump" Source="NET" Icon="Icons/Pump.png" Channels="1" Freq="1" Samples="1" >
      <Chan Id="0" Name="PumpOut" Units="rpm" />
    </OutputPacket>
  </GroupObject>
  <GroupObject Name="Blender Control" Icon="Icons/Blender.png" PW="1024" PH="575" >
    <InputPacket Module="2" Name="Net_CDI" Source="NET" Icon="Icons/cdi.jpg"
      Channels="15" Freq="1" Samples="1" TriggerFreq="0.16" >
      <Chan Id="0" Name="cPO2a" Units="mmHg" />
      <Chan Id="1" Name="cPO2v" Units="mmHg" />
      <Chan Id="2" Name="cSpO2a" Units="%" />
      <Chan Id="3" Name="cSpO2v" Units="%" />
      <Chan Id="4" Name="cPCO2a" Units="mmHg" />
      <Chan Id="5" Name="cPCO2v" Units="mmHg" />
      <Chan Id="6" Name="cpHa" Units=" " />
      <Chan Id="7" Name="cpHv" Units=" " />
      <Chan Id="8" Name="cHcrit" Units="%" />
      <Chan Id="9" Name="cHb" Units="%" />
      <Chan Id="10" Name="cFiO2Out" Units="%" />
      <Chan Id="11" Name="cGasFlowOut" Units="l/m" />
    </InputPacket>
  </GroupObject>

```

```

40     <Chan Id="12" Name="cVoiOut" Units="l/m" />
    <Chan Id="13" Name="cvct" Units="l/m" />
    <Chan Id="14" Name="cvdt" Units="l/m" />
  </InputPacket>
  <OutputPacket Module="12" Name="NetOutBlender" Icon="Icons/Blender.png" Source="NET" Channels="2" Freq="1"
    Samples="1" EnablePort="GeneralEnable" >
    <Chan Id="0" Name="FiO2Out" Units="%" />
    <Chan Id="1" Name="GasFlowOut" Units="l/min" />
  </OutputPacket>
</GroupObject>
<GroupObject Name="Meds Control" Icon="Icons/Meds.jpg" PM="1024" PH="575" >
  <OutputPacket Module="13" Name="NetOutMeds" Icon="Icons/Meds.jpg" Source="NET" Channels="3" Freq="1" Samples="1" >
    <Chan Id="0" Name="VolOut" Units="l/min" />
    <Chan Id="1" Name="ISDN" Units="%" />
    <Chan Id="2" Name="NEP" Units="%" />
  </OutputPacket>
</GroupObject>
<Calculate Name="NetTrigger" Trigger="NetInADC" >
  <Object Name="PumpGasCorrection" />
  <Object Name="ScopeA" />
  <Object Name="MemMeds" />
60 </Calculate>
</MDC>

```

