Assessment of goal-directed closed-loop management in intensive care medicine

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard-Karls-Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

Jörg Peter

aus Sigmaringen

Tübingen
2018

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	15.06.2018
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Wolfgang Rosenstiel
2. Berichterstatter:	PD Dr. Martin Schenk

Acknowledgements

First and foremost, I would like to thank Prof. Dr. Wolfgang Rosenstiel and PD Dr. Martin Schenk for providing the opportunity to write this thesis. Additionally, I would like to thank PD Dr. Martin Schenk for his confidence and the given liberties for developments within the experimental medical setting, including hardware modifications of medical devices.

Further, I would like to thank Dr. Wilfried Klingert and Dr. Kathrin Klingert as well as all other physicians, technicians, and medical students for sharing their medical expertise and the great time through days and nights while supervising the animal subjects during the conducted studies.

A special thanks goes to Philipp Gütschow for sharing his expertise regarding setup and configuration of the implemented and used infrastructure with various servers and managed switches.

Another thanks goes to all my friends and my colleagues at the computer science department for many inspiring conversations regarding scientific problems and an enjoyable time within and beyond the academic sphere.

Last but not least, I would like to thank my parents Isolde and Bernd Peter, who always support and encourage me.

Abstract

Given an aging population, shortage of nursing staff and a continuously increasing workload, automation in the medical sector is an important aspect of future intensive care. Although automation and machine learning are current research topics, progress is still very limited in comparison to other application areas. Probably one of the most serious problems is data shortage in a heterogeneous landscape of medical devices with limited interfaces and various protocols. In addition, the recording of data or, even more so, the evaluation of automation is limited by a complex legal framework. Given these complications and the sensitive legal nature of medical records, only very limited data is accessible for further analysis and development of automated systems.

For this reason, within the context of this thesis various solutions for data acquisition and automation were developed and evaluated concomitant to two clinical studies utilizing a large animal model in a realistic intensive care setting at the University Hospital Tübingen. Foremost, to overcome the problems of data availability and interconnection of medical devices, a software framework for data collection and remote control using a client-server architecture was developed and significant amounts of research data could be collected in a central database. Furthermore, a closed-loop controller based on fuzzy logic was developed and used for management of end-tital CO_2, glucose, and other parameters to stabilize the animal subjects during therapy and reduce caregivers' workload.

In addition to the fuzzy controller, closed-loop management for temperature and anti-coagulation could be established by developing hardware interfaces for a forced-air warming unit and a point-of-care analysis device, respectively. Besides further reduction of caregivers' workload, such systems can provide additional patient safety and allow management in settings where human supervision may not be present at all times.

One general and encountered problem for closed-loop control in a medical setting is limited availability of measurements, especially if manual blood withdrawals are required. As an initial step to address this problem, measured parameters from other devices as potential surrogates were evaluated in a comparison between different regression approaches. The required training data, a matched set of blood gas and monitoring parameters, was obtained by utilizing a developed algorithm for automated detection of withdrawal events.

Yet, besides any specific implementations and analysis, many general aspects regarding the physical implementation of such a system and interaction with caregivers could be evaluated in the experimental setting and might guide further development of clinical automation.

Kurzfassung

Angesichts der alternden Bevölkerung, des Mangels an Pflegekräften und der ständig steigenden Arbeitsbelastung ist Automatisierung ein wichtiger Aspekt zukünftiger Intensivmedizin. Obwohl Automatisierung und maschinelles Lernen aktuelle Forschungsthemen sind, ist der Fortschritt im Vergleich zu anderen Anwendungsbereichen jedoch noch sehr begrenzt. Eines der größten Probleme ist wohl die Datenknappheit in einer heterogenen Medizinproduktelandschaft mit begrenzten Schnittstellen und zahlreichen unterschiedlichen Protokollen. Darüber hinaus sind die Datenerfassung und erst recht die Erprobung einer Automatisierung durch ein komplexes rechtliches Rahmenwerk eingeschränkt. Aufgrund dieser Komplikationen und der sensiblen Rechtslage für Patientendaten sind diese nur sehr begrenzt für weitere Analysen und die Entwicklung automatisierter Systeme zugänglich.
Im Rahmen dieser Dissertation wurden daher verschiedene Lösungen zur Datenerfassung und Automatisierung begleitend zu zwei klinischen Studien des Universitätsklinikums Tübingen am Großtiermodell in einer realitätsnahen Intensivstation entwickelt und evaluiert.
Um die Probleme der Datenverfügbarkeit und Vernetzung medizinischer Geräte zu lösen, wurde vorrangig ein Software-Framework für die Datenerfassung und Steuerung mittels einer Client-Server-Architektur entwickelt und umfangreiche Forschungsdaten in einer zentralen Datenbank gesammelt. Darüber hinaus wurde ein auf Fuzzy-Logik basierender Regler entwickelt, welcher zur Stabilisierung des endtitalen CO_2, Glukose und anderen Parametern verwendet wurde und damit die Arbeitsbelastung der Pflegekräfte reduzieren konnte.
Zusätzlich zum Fuzzy-Regler konnten durch die Entwicklung von Hardware-Schnittstellen für Geräte zum Temperaturmanagement mittels luftbasierter Wärmedecken und zur Messung der Blutgerinnung geschlossene Regelkreise aufgebaut werden. Neben einer weiteren Arbeitserleichterung für die Pflegekräfte können solche Systeme zusätzliche Sicherheit für den Patienten bieten und die Anwendung in nicht ständig überwachten Bereichen ermöglichen.
Ein allgemeines und auch beobachtetes Problem für Regelkreise im medizinischen Bereich ist die begrenzte Verfügbarkeit von Messwerten, insbesondere bei manuellen Blutentnahmen. Als erster Schritt zur Lösung dieses Problems wurden Messparameter anderer Geräte als potentielle Ersatzparameter mit verschiedenen Regressionsansätzen analysiert und verglichen. Die dazu erforderlichen Trainingsdaten, Paare von Blutgas- und weiteren Vitaldaten, wurden mit Hilfe eines entwickelten Algorithmus zur automatisierten Erkennung von Blutentnahmen erzeugt.
Abgesehen von diesen konkreten Anwendungen und Analysen konnten in der experimentellen Evaluation auch viele generelle Aspekte der realen Implementierung eines solchen Systems und die Interaktion mit Ärzten und Pflegekräften untersucht werden und damit der Entwicklung weiterer klinischen Automatisierung dienen.

Contents

List of Abbreviations

ABP arterial blood pressure
ACT activated clotting time
AEP auditory-evoked potentials
ANN artificial neural network
ANSI American National Standards Institute
AP Autopilot
API application programming interface
AWMF Arbeitsgemeinschaft der Wissenschaftlichen Medizinischen Fachgemeinschaften
BIS bispectral index
BGA blood gas analysis
BP blood pressure
CCM critical care medicine
CDSS clinical decision support system
CGM continuous glucose monitoring
CIS clinical information systems
CNS central nervous system
CO cardiac output
CPR clinical patient record
CSV comma-separated values
CVC central venous catheter
CVP central venous pressure
DHCP Dynamic Host Configuration Protocol
DL German Landrace
DNS Domain Name System
ECG electrocardiogram
EHR electronic health record
$etCO_2$ end-tidal CO_2
FDA U.S. Food and Drug Administration
FL Fuzzy Logic
GA general anesthesia
GPIO general purpose input/output
GUI graphical user interface
HDF hierarchical data format

HIS	hospital information systems
HL7	Health Level Seven
HR	heart rate
ICU	Intensive Care Unit
ICP	intercranial pressure
ISO	International Organization for Standardization
IT	information technologies
IU	International Unit
IV	intravenous
KCl	potassium chloride
LAG	Link Aggregation Group
MAR	Missing At Random
MCAR	Missing Completely At Random
MV	minute volume
NaBic	sodium bicarbonate
NMAR	Not Missing At Random
NTP	Network Time Protocol
OR	Operating Room
PEEP	Positive End-Expiratory Pressure
PID	proportional–integral–derivative
PIP	Peak Inspiratory Pressure
POC	point-of-care
ROS	robot operating system
RR	respiratory rate
STP	Spanning Tree Protocol
SV	stroke volume
SVR	Support Vector Regression
SVV	stroke volume variation
TGC	tight glycaemic control
TICoMS	Tübinger ICU Control and Monitoring System
TIVA	total intravenous anesthesia
TTC	time to criticality
TTM	targeted temperature management
TV	tidal volume
UKT	University Hospital Tübingen
VLAN	Virtual Local Area Network
VNA	Volume Need Analysis
XML	Extensible Markup Language

1 Introduction

Would you entrust your life to a computer while staying critically ill in a hospital? One might hesitate to answer such a direct question with yes, but everybody is more or less unknowingly entrusting their lives to machines every day to a varying and even greater extent. Would one hesitate in an equal fashion if asked to travel by plane? In this case, most people would agree and not even contemplate too much about such a common situation. But is this situation really that different? Sitting in an enclosed space, several thousand feet above the ground with no way to exit. One might argue that the pilots in the cockpit are flying the plane, but most of the time, except for the start and landing, the plane is operated by computers controlling countless functions of electronic devices and navigating with autopilots, only guided and instructed by the navigation and control input from the pilots. Is this really so much different from a computer performing therapeutic actions in a medical scenario, with physicians guiding the therapy? — But whereas automation and computerization found appeal in most of our life situations, the medical sector is still an area where people mostly trust the experience of another, professionally trained, human. But can a physician always stay at your side, observe every little change in your condition and immediately react to it? With no human being perfect and an aging society, the workload and burden for caregivers is steadily increasing. Computer-aided decision making and automated therapy guidance therefore will be an important aspect of critical care in the future. Many questions covering various aspects concerning closed-loop systems for goal-directed therapies, machine-learning, and decision making based on collected clinical data are subject to basic research. Yet, fundamental technical aspects like the interconnection and communication between different medical devices, data storage, and processing are unsolved. Thus, in clinical practice, manual input and processing of data observed at another medical device is still required on a daily basis. In the course of this thesis, these aspects will be covered, providing experimental solutions for data collection and control systems as well as first analyses based on collected data.

1.1 Motivation

Research and advantages in clinical techniques and equipment allow for treatment of more and more complex and life-threatening conditions [1]. Yet, most knowledge and decision making still remains empirical [2] and understanding of physiological models and parameter interactions is still limited. Whereas in other sectors, information technology and automation already play an essential role for reduction of workload and increased safety, within medicine and critical care, those advantages are still mostly limited to patient management and guidance systems [3] and still heavily rely on human action and expertise.
Despite an advancing technical progress, hospitalized care still involves countless manual and repetitive tasks like manual calculations, and entering numbers and measurements to various devices, instead of being able to focus on therapeutic tasks that benefit from human knowledge, expertise and flexibility. When automation or closed-loop systems are available in the medical sector, they often only are present in a less obvious scale. Such application include automated management of gas concentrations in medical ventilators or calculations within patient monitors.
Additionally, whereas an increasing number of medical devices in intensive care settings is available, those systems often have made patient assessment and care even more complex and resulted in an increased workload [1]. Nowadays, many point-of-care (POC) and laboratory systems, e.g., for assessment of blood gasses of blood clotting time are readily available. While aiding high-level critical care, manual workload and critical thinking is often required for correctly handling those tools and resources by physicians and nursing staff, instead of being part of clinical routine [1].
In general, assessment of critical situations and notification of caregivers is still based on simple threshold alarms. Given such a broad variety of medical devices monitoring the individual patient, this results in high false alarm rates and additionally increased workload for caregivers. Comparison of measurements from different sources could lead to major improvements as such a comparison of different vital parameters would allow to determine if a sensor, like a clip-on sensor for oxygen saturation just has fallen off, while related measurements are still present from other sources, or if in fact a critical condition requiring imminent action and notification to caregivers is present.
In line with the concept of simple threshold alarms, many aspects of care are still based on simple therapeutic protocols, check-lists, and guidelines. This protocolized care may be useful in situations with limited staff but cannot provide the high quality of care achievable by well-trained ICU physicians and nurses [3]. Whereas for usage by caregivers, such tools are essential for fast reactions, reduction of workload, and to provide a basic level of care, implementation of those protocols is just a first and initial step for automated systems. For personalized medicine of the future, therapeutic decision making must be performed on individual patient basis [3]. Thus, more advanced systems utilizing machine learning for information gathering from a large data base of medical knowledge and adaption to the individual patient are required.
Along with the trend of an aging population, a rising demand for critical care and a limited number of skilled clinicians, problems like understaffing and providing adequate care for all patients may become more severe in the future [3, 4]. By application of information

technologies (IT) to this sector, there is a large potential for patient management to be revolutionized [3] as the trend of computerized medicine has just started within the last decade and developed into a growing research field. Medical device software already plays an increasingly important role in healthcare [5] and research provides tools for better patient care and access to patient resources [6]. As automated systems do not require to share their workload between different patients, they may provide significant advantages in therapeutic adjustments. This may include the detection and assessment of critical situations more easily, thus allowing more proactive interventions [3].
As the Intensive Care Unit (ICU) setting is used for treatment and monitoring of possibly life-threatening conditions and stabilization of critically ill patients, many technically advanced medical devices and a high level of monitoring is already present, providing a unique opportunity for revolutionizing critical care [7]. Moreover, given that the ICU is one of the most resource-intensive and expensive hospital areas, further research and technological advantages for reduction of workload and aiding the delivery of evidence-based, cost-effective care that may lead to better patient outcome and reduction of time of stay should actually be a well-established research field even from an economic standpoint [8]. Yet, whereas the great potential of automation and closed-loop systems for improving care can clearly be seen, there are still many obstacles and difficulties on the way towards automated systems.

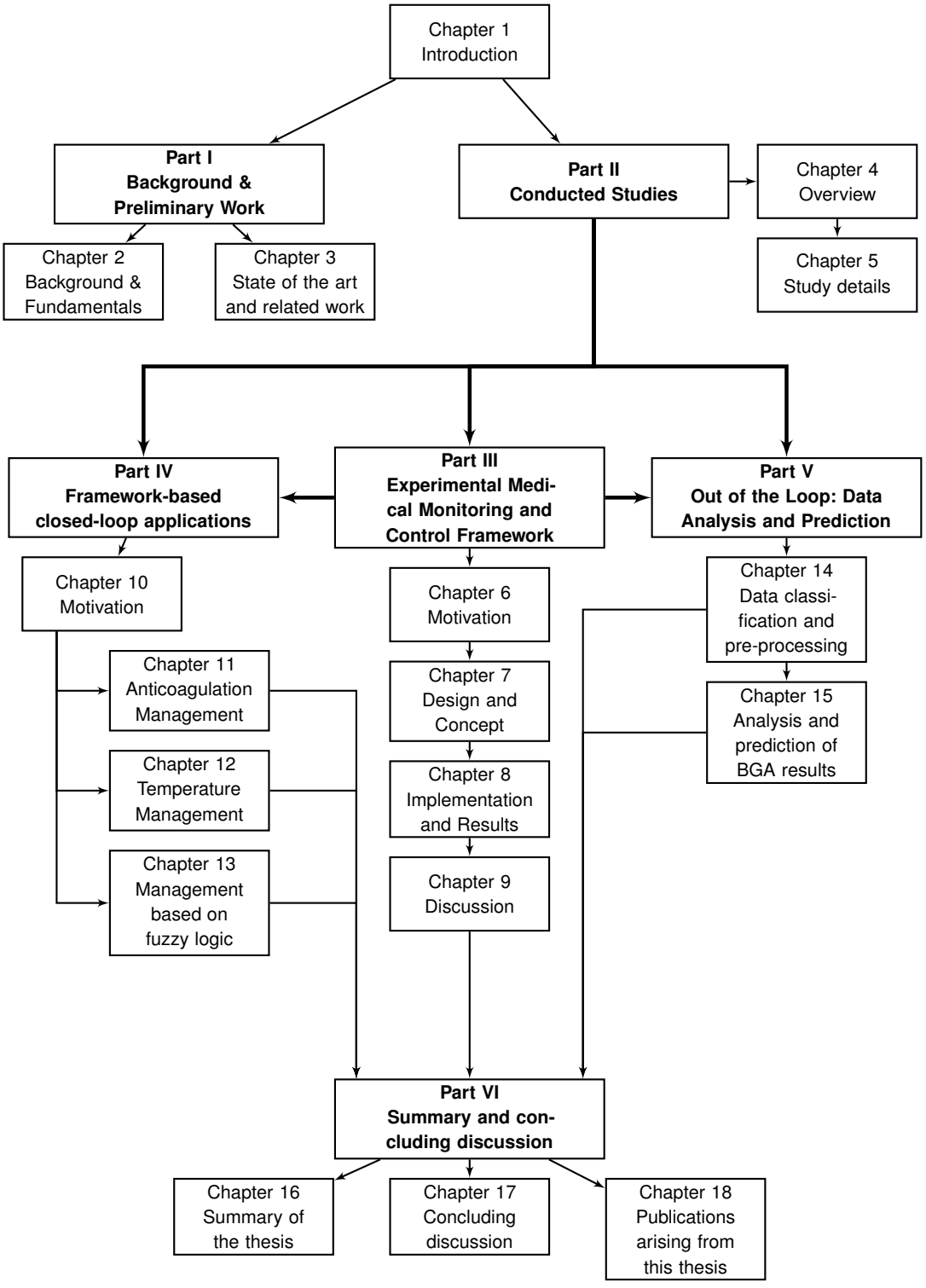
An important difference to many other industrial and research fields, mainly driven by the laws of physics and mastered by skilled engineers, is the still present lack of knowledge regarding the human body and life itself. One difficulty for application of any automation is the system to be controlled: the human body as a very intricate and multifaceted system [7] that cannot easily be described by biochemical, physical, or other models. Thus, especially given such a complex system like the human body, a comprehensive base of knowledge and interactions for various medical conditions and demographics as a digital foundation for development of better models and algorithms is essential.
Unfortunately, whereas more and more medical devices are available to collect a multitude of patient information, displayed on the numerous medical devices, especially in the technologically advanced ICU [7], the majority of displayed information is forever lost as it cannot be collected and be made available for further processing [9]. This is caused by the historically independent development and implementation of such systems and medical devices, additionally performed without a strong focus on further data processing and thus still lacking standardization and interoperability among them [7, 10, 11].
This problem needs to be addressed as an uttermost important first step as high-resolution data from a broad variety of vital parameters and devices is a requirement for any further processing. Once such information is obtainable and gathered, machine learning and data mining can be used to transform the collected information into useful data for better diagnosis, event detection and decision making [12].
Besides those major technical obstacles still present, the problem of data abundance must be considered from a legal perspective, too. Many aspects regarding patient safety and the certification of medical devices and computer systems that might be able to collect such data in the first place are a further limiting factor. And additionally, once data can be collected,

legal limitations considering patient rights and privacy aspects of clinical patient data as well as many unresolved questions regarding liabilities by using automated systems, even beyond the scope of clinical decision making, must be addressed.
Another mainly non-technical but not less important aspect needs to be considered when dealing with the idea of automated systems and closed-loop therapies: Acceptance of such novel approaches might be highly impaired by generating a feeling or even fear of being replaced in caregivers. Additionally, given an automated system, even including patient-specific adaptive algorithms, applied to each patient in the same way, a contradiction between such a generalized approach and the philosophy of individualized care to best of knowledge by physicians and caregivers may arise on first sight. To achieve acceptance for this novel technology, such aspects should be considered and addressed, providing a helpful tool aiding the caregivers, still allowing flexibilities and adaptions, but not forcing them to decisions that might contradict their expertise and critical thinking [1].
Yet, despite all those limiting factors, obtaining a substantial data base and practical evaluation of different approaches are essential first steps for further advantages regarding the reduction of workload and application of machine-learning and closed-loop systems for better patient care in the clinical environment that need to be performed for developing the future of medicine and critical care. Furthermore, such fundamental first steps might allow to evaluate approaches that will help to establish concepts accepted by the medical community and provide a foundation for legal considerations outlining possibilities and risks that need to be weighted against each other.
Given the outlined significant legal obstacles regarding development of new medical devices and collection of patient data, one way of collecting knowledge applicable to humans is by using appropriate animal models in a first step. With such studies, fundamental knowledge regarding automated systems and the interaction with the caregivers may be developed and evaluated prior to further clinical trials. However, such studies should always be performed in a way to gather and preserve all available information and knowledge, thus limiting animal harm and the required number of animal subjects. Therefore, instead of directly developing individual therapeutic systems and concepts with limited and often still manually performed data collection for only few required parameters, fundamental questions regarding the future of medicine with the interconnection of medical devices should as well be solved for studies involving animals in such a first step. Once given information stored in a central, digital database, this information may then be used for further analysis and the development of advanced models that may improve patient care and reduce the need for further animal studies as the already gathered knowledge might be usable and sufficient for the evaluation of other or more in-depth research questions asked in the future.

1.2 Outline of the thesis

The thesis is divided in several parts. Starting with Part I, a general introduction to the relevant medical and computer science backgrounds is given. Thereafter, an overview of relevant and related research with a summary of the state of the art is presented.
Next, an overview of the conducted clinical studies performed at the University Hospital Tübingen (UKT) is presented in Part II. As the presented studies were not especially designed or solely carried out for the scope of this thesis but instead are part of ongoing clinical research, those study details are presented as a background chapter to provide the available clinical framework and setting. The conducted studies provided the working base for the development of different automated solutions and applications described in subsequent parts of this thesis. Yet, development of the framework and automated solutions was of course interlinked with the progress of the clinical studies to improve and stabilize study conditions and enhance therapeutic treatments to obtainable more reliable results for clinical research.
For collection of data in the used experimental clinical setting, a framework for device interaction and information gathering has been developed as an essential part of this thesis and provided the foundation for closed-loop applications and machine learning. Part III of this thesis describes the framework's technical aspects with its requirements, functional concepts, design, and implementation. Furthermore, the results of the successful application of the framework during the conducted clinical studies are presented from a technical, framework-based perspective, including the number and variety of collected parameters and the framework performance as a general overview without a clinical focus or going into detail of specific applications, which are described in subsequent chapters.
Following the general description of the framework, Part IV presents its clinical application for specific tasks within the experimental study context, consisting of individual closed-loop systems presented with their clinical motivation, used methods, implementation, and the obtained results. Given the clinical context of the conducted studies, the presented applications were chosen to stabilize the study conditions for the clinical research questions by improving maintenance of hemo- and homeostasis and solving practical problems to reduce workload and human error potentials.
Besides the closed-loop systems directly interacting with the animal subjects and influencing homeostasis, retrospective analysis of the collected data was performed and is presented in Part V. As closed-loop systems based on measured blood gas parameters were relevant for homeostasis management, prolonged measurement intervals provided a major limitation for the adaption frequency. Therefore, retrospective analysis on the collected parameters focused on the automated detection of blood withdrawal timepoints to reduce the influence of measurement delays. Given a matched set of parameters within the time domain, this allowed for the evaluation of surrogate parameters that might allow a faster adaption frequency of the closed-loop systems by including more frequently available parameters from patient monitoring instead of relying solely on the BGA results, thus potentially improving homeostasis management.
Finally, this thesis concludes with an overall discussion about automated intensive care systems and an outlook for future research with possible applications in Part VI.

An overview of the individual parts and chapters with their relations is shown below.



Part I

Background & Preliminary Work

2 Background & Fundamentals

To provide a fundamental understanding for readers of different backgrounds, a brief introduction to the relevant basics of the medical and computer science domains should be given. First, the clinical background with its terminology, the overall clinical setting with the used devices and a general understanding of human physiology and therapeutic measurements is provided. Then, a brief overview of the fundamentals of animal studies and the porcine surrogate model with a comparison of the most important vital parameters to the human physiology is given. Subsequently, the computer science background with general definitions and fundamentals, fuzzy logic and methods of statistics and machine learning is outlined.

2.1 Medical Background

In this first section of the introduction, a brief overview of the medical background and terminology used within this thesis, including introductions to the medical setting, devices, drugs, and study model will be given. However, besides technical and physiological aspects, another important aspect needs to be considered and should become clear alongside the introduction of devices and procedures: The medical staff, caring for the critically ill patients, often having several responsibilities and high workload.

2.1.1 Hospital Setting

Within the hospital, Operating Rooms (ORs) and Intensive Care Units (ICUs) are settings for dealing with patients in critical conditions. Whereas the OR is used for surgical interventions and procedures, the ICU is used for life-saving treatments and stabilization of acutely ill patients using technologically advanced systems [7]. Evolved by the increased need for intensive monitoring of such ill patients and improved patient outcomes, close monitoring and treatment in specialized units for critical care medicine (CCM) has been established since the first quarter of the 20th century, inter alia due to polio epidemics [7, 8, 13]. ICUs in the US alone account for roughly $80 billion dollars, over 13% of total hospital costs or about 4% of US health expenditures [7, 14]. Such a high level of patient care can only be achieved by a broad variety of advanced medical devices and procedures, which are steadily improved by ongoing research allowing the treatment of increasingly ill patients [8]. As those devices are an important aspect of care, a brief introduction to those technologies, devices and procedures is given in the following sections.

2.1.2 Medical Devices and Systems

In the Operating Room (OR) and Intensive Care Unit (ICU) settings, a broad variety of special devices, some illustrated in Figure 2.1, is required to provide an adequate support for critically patients during their surgical interventions or the recovery phase. For the U.S. market alone, over 200,000 medical devices were registered by the U.S. Food and Drug Administration (FDA) in 2007 [10]. Whereas most medical devices provide interfaces for data access and export, development was most often performed independently with a lack of standardization. Thus, interoperability of those devices is limited [7, 10]. Hence, the large volume of data that can be collected from those devices cannot be thought of as an entity but a cluster of individual data management and recording system in a heterogeneous landscape [7]. Integration of this data is a costly and non-trivial process, requiring well-defined standards among many devices of competitive manufacturers [7]. Besides the data collection and monitoring, experimental devices might be equipped with additional remote-control capabilities, exceeding the current legal limits for such medical products and thus only usable in experimental settings. To provide a general understanding of the devices and procedures, mentioned and used in the further course of this thesis, this section aims at giving a brief overview of the most important devices for patient monitoring, breathing-support, drug infusions and laboratory analysis.

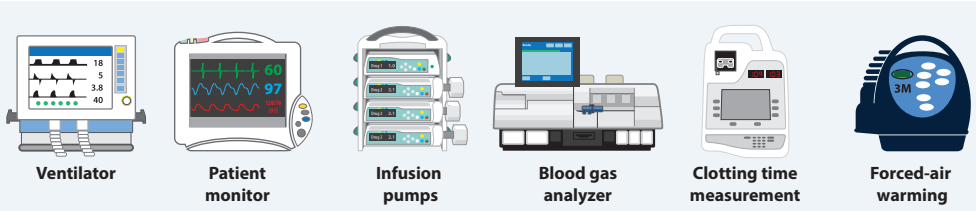


Figure 2.1: Illustration of different medical devices used for patient monitoring, laboratory analysis and therapeutic managements.

2.1.2.1 Patient monitoring

Patient monitoring is required to observe patient's vital signs with the help of specialized devices, collecting and presenting the measured information on a bedside screen. The most generally known device is a patient monitor used to monitor vital signs like oxygen saturation, heart rate, blood pressure, electrocardiogram (ECG), temperature, and potentially many other more specialized parameters. The obtainable parameters are highly dependent on the individual patient's monitoring requirement and the availability of equipment within the hospital ward. The most essential measurements like the blood oxygen saturation and hearth rate were already establish in ICUs in the 1980s [7]. Nowadays, ECG is a standard measure but further analysis of the heart's performance, complexes and characteristics within the measured wave depend on the features of specific device [7]. To obtain information about arterial blood pressure (ABP) and central venous pressure (CVP), specialized catheters are used. Those catheters are inserted in central arterial and venous blood vessels, respectively, and connected to pressurized fluid bags with in-line pressure sensors to obtain measurements of a standing fluid pillar that can be displayed on the patient monitor.

2.1.2.2 Ventilation devices

Mechanical ventilation devices are used for artificial ventilation of a patient's lung if they cannot breathe naturally [15]. This, for example, is the case in a fully sedated patient, where muscle relaxants prohibit voluntarily movements including breathing efforts. To deliver the required air or even pure oxygen and ventilate the lungs, ventilators are most commonly connected to a tracheal tube, inserted through the patient's mouth to provide access to the airways.
In ICUs, ventilators were one of the first established support devices [13]. Whereas ventilation was initially based solely on pneumatic or gas driven processes, developments during almost 100 years have led to sophisticated designs. By utilizing embedded computers and microprocessors to electronically control servo valves and transducers and for measurement of various volumes and pressures among many other parameters, device performance and settings are constantly monitored to maintain the desired state and report any errors or malfunctions [7, 15].
Breathing itself consists mainly of two phases: an inspiratory phase and an expiratory phase, whereas in the first phase, the lungs are inflated and in the second one deflated. The amount of air moved during such a breath is called the tidal volume (TV) [15]. The number of breaths observed within a minute is called the respiratory rate (RR). By multiplication of this rate with the TV, the minute volume (MV) can be calculated [15]. During general anesthesia and other conditions, where no self-breathing of the patient is possible, mechanical ventilation is performed in a mandatory mode, forcing the patient to breath in a specific way. To prevent the lung's alveoli from collapsing during the expiratory phase, a Positive End-Expiratory Pressure (PEEP), can be used to retain a therapist-defined pressure level within the lungs at the end of expiration cycle [15].
Included within the long and differentiated developments of ventilation support, two rather distinct modes for mandatory breathing evolved: The first, pressure controlled ventilation, is a mandatory mode giving rise to a defined inspiratory airway pressure. The second, volume controlled ventilation, delivers a specific TV during inspiratory phase. Both modes try to achieve the desired pressure or volume independently of patient's airway resistance or other restricting factors, respectively [15, 16].

2.1.2.3 Infusion Pumps

Infusion pumps are used for a controlled administering of drugs. As many different medications may be needed, multiple infusion pumps are often arranged in pillars. Administering of the infusions to the patient is performed through an intravenous (IV) access with defined infusion rate sets at the individual infusion pumps. This intravenous (IV) administering allows a rapid distribution of substances and continuous drug therapy through the blood stream. As the circulatory system is the body's primary pathway for distribution and exchange, providing a complete circulation within 60 seconds in healthy adults, IV administering become the mode of choice in clinical care. To perform the infusion, pumps generate an active flow to push fluids through the IV access into the patient's blood stream. This is performed with high accurate over a wide range of infusion rates that can reach up to 1000 $\mathrm{mL/h}$. Mainly, two specific types of infusion pumps can be distinguished: Pumps using infusion lines to connect to bags or bottles and pumps with inserted syringes. When using infusion lines

for connection to fluid reservoirs, sensors for counting fluid drops passing through a drop chamber are often used for automated recalibration and monitoring of the desired flow rate. To move the fluid within the infusion line and provide positive pressure, linear peristaltic pumps using an array of cam-driven actuators are used on a short segment of the infusion line to provide a sinusoidal wave motion, thus forcing the fluid toward the patient's IV access. Syringe pumps employ a syringe as a fluid reservoir and administer its content by linear advancement of the syringe plunger with a motor and precision gears [15].

2.1.2.4 Blood gas analyzer

The blood gas analyzer is a device used to perform a BGA by measuring the composition of blood samples using a set of electrodes and other measurement units. It utilizes solutions with defined compositions as well as different gases for measurement and flushing. Analysis itself is performed by transportation of the aspirated blood sample through an arrangement of tubes, passing sensor arrays of measurement electrodes.

2.1.2.5 PiCCO system and measurement

The stroke volume (SV) and cardiac output (CO) are important cardiological measurement obtainable by using specific aortic flow probes. Based on this measurement, the difference of SV between inspiratory and expiratory ventilation phase, the stroke volume variation (SVV), as an important hemodynamic parameter can be calculated. To replace the need for such a specific invasive sensor, systems using a pulse contour analysis and calculation of SVV and CO were developed [17]. One such a well-established system in Europe is the PiCCO technology for expanded hemodynamic monitoring [16]. Measurement is performed with a central venous catheter (CVC) and an arterial catheter. Both catheters are equipped with thermistors, measuring the blood temperature for analysis with transpulmonary thermodilution. Therefore, a defined amount of cooled infusion fluid can be injected into the CVC, resulting in a temperature change at the venous and arterial thermistor, whereas the measured delay in temperature changes is used for calculation of a continuous CO [16]. Besides the measurement of the CO, additional parameters and continuous estimations, dependent on the device's specific algorithms, can be obtained.

2.1.2.6 Clinical systems and information storage

Besides clinical measurement devices, a variety of different systems is used for management of patient data within the hospital setting. Management of administrative data from patients, like demographic, location or billing information from several departments is normally stored using hospital information systems (HIS) [15]. On contrary, clinical information systems (CIS) are used for management of the clinical and nursing needs of the patient [15]. Beside those two, more technical and administrative systems, the clinical patient record (CPR), also called electronic health record (EHR), is used for storing the patient's vital data and measurements [15]. Digital data within the CPR can then be used by clinical decision support systems (CDSSs), which are computer-systems aiding physicians' decision making [15].

2.1.2.7 Legal framework & clinical requirements

Given the technical progress in other industrial fields and the great potential for improvement in healthcare, the development of novel medical devices with more advanced features and improved therapies is an ongoing task and challenge. Yet, technical possibilities and advantages are only one side of the coin during development. Due to high safety requirements and responsibilities of the individual devices, a large legal framework shapes the other side. Of course, even used software and algorithms are often to be classified as a medical device and therefore must satisfy the specific regulations regarding their application and functional characteristics. This classification is graded by assessment of the risk for human harm, providing a variety of applicable restrictions regarding the risk level [6]. Explicitly stated by current European legislation, electronic systems and software shall be designed to ensure repeatability, reliability and performance [18, p. 17.1.]. Therefore, the general functionality of all parts, formulation and composition, namely including the software, shall be represented by diagrams and drawings, to aid understanding of the underlying mechanism [18, Annex 2 1.1(j)]. Given this legal framework, many advanced machine learning approaches cannot directly be applied to the medical sector, as the obtained results and knowledge is hidden within a *black box* and often cannot be comprehended to fulfill such legal requirements.

2.1.3 Patient management

Within the given critical care environment and the presented essential devices, critical care for the patient can be provided. This care can be categorized in different aspects, which are briefly introduced to provide the essential context for aspects of automation regarding hemo- and homeostasis management to stabilize the animal subjects that will be presented in further parts of this thesis.

2.1.3.1 Anaesthesia management

Anaesthesia can be provided as local or general anesthesia (GA). Whereas local anaesthesia is used for pain reduction (analgesia) on specific regions on the body, GA is further used to induce patient's unconscious (hypnosis) and loss of memory (amnesia) in a reversible, controlled manner [8]. This is achieved through special anaesthetic drugs binding to specific receptors throughout the brain, brainstem, and spinal cord and thus interfering with neural signal processing [8]. Such anaesthetic agents can be administered through gas inhalation or as total intravenous anesthesia (TIVA) with a combination of drugs like fentanyl, midazolam and ketamine. Beside patient's unconscious, general anesthesia is used to induce immobility by muscle paralysis to inhibit patient's response during surgical procedures [8]. This inhibition of voluntary and involuntary body response is also an important factor for establishing mechanical ventilation with endotracheal intubation [8]. However, by inhibiting pain and muscular responses, patient's brainstem may still react to stimuli with increased ABP, heart rate (HR) and RR [8]. Therefore, monitoring of the patient's HR, RR, pressures and oxygen saturation is mandatory during GA [8].

2.1.3.2 Respiratory management

Respiratory management is the task to maintain gas exchange and avert critical conditions. Those may include, hypo- and hypercapnia, with critically decreased and increased CO_2

levels in the blood gases, respectively, or similar conditions regarding the oxygen saturation (hypo- and hyperoxemia). This goal can inter alia be achieved by adaptions of the oxygen saturation or the MV, depending on the ventilation mode defined by RR, PEEP and Peak Inspiratory Pressure (PIP). An important aspect of management is to keep the lungs inflated and avoid collapsing by maintaining positive pressures. Yet, increased pressures should be avoided as they can damage the lungs and impede weaning from mechanical ventilation.

2.1.3.3 Hemodynamic management

Hemodynamic management is the process of maintaining stable conditions regarding the blood circulation by hemodynamic monitoring. This involves observation and control of blood pressure (BP), e.g., to avoid hypo- and hypertension, and monitoring of CO and the overall intravascular fluid content to detect and avoid hyper- or hypovolemia.

2.1.3.4 Homeostasis management

For management of the balance of the biochemical body functions, homeostasis, a broad variety of parameters needs to be observed and kept within the physiological ranges. The blood pH value is one of the most important parameters to be kept within a nominal range of 7.35 to 7.45 [8]. Only within this range, essential vital functions like metabolism and oxygen transport for cellular perfusion can be maintained [8]. Acid-base balance and the resulting pH value is challenged by a broad variety of metabolic processes altering the acid and alkali composition within the body. However, a healthy body is able to compensate this fluctuations by means like buffering hydrogen ions (H^+) in the blood and elimination thereof through CO_2 exhalation by the lungs through adaption of RR and MV or excretion within urine by the kidneys [8, 16]. For this buffering system to work, electrolytes like calcium, chloride, magnesium, potassium and sodium are essential and need to be maintained within defined ranges. Restriction of the body's abilities to achieve this goal, often caused by respiratory or metabolic complications lead to conditions known as acidosis and alkalosis, resulting in a decreased (acidemia) or increased (alkalemia) pH, respectively [8]. In critical care, were medical conditions often substantially prohibit the body's own abilities, recovery and maintaining a balanced state is in general aided by periodical BGA and administering a variety of IV buffering solutions with additives of the required electrolytes. Thus, considering automated solutions for homeostasis management, these different buffering solutions play an important role. A brief overview of the individual electrolytes and metabolic conditions is given in the following sections.

2.1.3.4.1 Sodium and natremia Sodium (Na^+) is one of the most important electrolytes. Disruptions of physiological levels caused by loss of water through the kidneys or thirst as well as administering of hypertonic solutions can lead to hyponateremia [8]. On contrary, significantly increased levels are termed hypernateremia.

2.1.3.4.2 Potassium and kalemia Perturbations in the extracellular potassium (K^+) levels may need to be treated to prevent conditions that may have lethal consequences. A significantly increased potassium level (hyperkalemia) may cause severe cardiac disorders [8]. On contrary, reduced levels of potassium (hypokalemia) may be the result of severe diuresis and may cause cardiac arrhythmia, too [8].

2.1.3.4.3 Calcium and calcemia Dissolved calcium (Ca^{2+}) is another important electrolyte that needs to be monitored to prevent cardiac complications. Increased levels of calcium (hypercalcemia) can be observed by a shortened QT interval in ECG measurements [8] and with BGA. Alkalosis may be caused by lack of calcium (hypocalcemia) and needs to be treated with calcium supplementation among other therapeutic measures [8].

2.1.3.4.4 Metabolic acidosis and alkalosis Changes in the measured pH caused by various changes in metabolic processes are called metabolic acidosis and alkalosis, leading to accumulation H^+ or a loss of HCO_3^- in acidosis or vise versa for alkalosis. Primary metabolic acidosis is characterized by an arterial pH < 7.35 and a concentration of HCO_3^- $< 22 mEq/L$. On contrary, alkalosis is diagnosed if pH > 7.45 and $HCO_3^- > 26 mEq/L$ [8]. Causes for alkalosis include hypokalemia, hypomagnesemia, chlorine deficit as well as extracellular volume depletion (hypovolemia), e.g., due to use of diuretics [8, 16]. Acidosis on the other hand may be caused by increased chlorine levels, hypoglycemia or other general causes that disturb the acid-base balance like hypovolemia. BGA is therefore an essential tool for correct assessment of the present condition and underlying causes.

2.1.3.4.5 Respiratory acidosis and alkalosis Critical changes in the pH level due to respiratory causes are called respiratory acidosis if the pH level of the blood is decreased or increased respectively and changes in the partial CO_2 pressure are observed. Respiratory acidosis is defined a pH < 7.35 and a measured partial CO_2 pressure level above 44 mmHg, whereas the alkalosis condition is defined as pH > 7.45 and CO_2 below 36 mmHg [8]. As an increased CO_2 level decreases the pH, the measured partial CO_2 pressure is a representation of the balance between the body's CO_2 production and exhalation [8]. Increased respiration rates are a cause for respiratory alkalosis, whereas low RRs or airway obstructions can lead to respiratory acidosis [8]. Management of RR and monitoring of the partial CO_2 pressures are therefore an important therapeutic aspect as a precaution and for treatment of these conditions.

2.1.3.5 Glucose management

The blood sugar level can be measured with BGA or other specific tests. In critically ill patients, injuries can introduce increase glucose levels (hyperglycemia) due to inflammatory or hormonal processes. This may increase the risks of infection and influence patient outcome. Target levels of 80 mg/dL–110 mg/dL are desired, but a drop in glucose levels below this range can easily result in life-threatening hypoglycemia. Therefore, in clinical practice safer levels of 140 mg/dL–180 mg/dL are often accepted as levels below 140 mg/dL require tight control and active management [8].

2.1.3.6 Anti-coagulation management

The balance of blood thickness and clotting characteristics is another important aspect of intensive care. In hospitalized patients, which are often bedded for an extensive period of time, the risk of thromboses or other clotting incidents is drastically increased. An effective method to reduce this risk is the management of clotting factors within the patient's blood. Unfortunately, whereas reducing the clotting risk is solving one serious medical complication, it is facilitating another: By reduction of the blood's ability to clot, the risk

of life-threatening internal bleeding is increased. Therefore, clotting factors cannot arbitrarily be reduced, but must be managed carefully as a tradeoff between the risks of internal bleeding and thrombosis. This can be performed by regular measurements of the current coagulation state and appropriate adaptions. Besides laboratory analysis taking a significant amount of time, assessment can be performed with significantly faster POC tests. Regarding to coagulation, measurement of activated clotting time (ACT) is the most commonly used bedside test, whereas the activated partial thrombin time (APTT) is a well-established laboratory analysis [1]. The ACT test, first described by Hattersley in 1966 [19], is a whole-blood coagulation test based on clotting through the activation of the intrinsic pathway of the coagulation cascade involving fibrinogen, platelets and red blood cells [1, 20]. Measurement is performed by adding a blood sample to a sample tube containing an activator agent to increases clotting speed and measuring the required time to achieve clotting [1]. Given the measured ACT results, therapeutic action can be taken by administering of unfractionated heparin to decrease clotting as the long standing standard drug for anticoagulation management [1, 21].

2.1.3.7 Temperature management

Management of patient temperature should be seen as an important part of the overall intensive care strategy. As a goal-directed approach, targeted temperature management (TTM) defined by a therapeutic temperature profile may be used [22]. Yet, temperature is often not regularly checked in perioperative settings [23] and, given the overall workload, only performed as a subsidiary task besides the other goals. Assessment of patient temperature can be performed by a variety of different methods, including measurements on the skin, oral, rectal or with temperature sensors included in catheters that are inserted into bladder or blood vessels [24, 25]. In general, a normal body temperature (normothermia) of 36.5 °C–37.5 °C [26] should be kept. If the body temperature exceeds this nominal range an increased body temperature (hyperthermia) due to fever or other responses, or hypothermia due to heat loss can be observed. This loss is caused by radiation, conduction, convection and evaporation [25]. Especially under GA, this auto-regulation of the body can be impaired. Drugs may have vasodilating effects, leading to a dilation of blood vessels and redistribution of the body heat to the surface, causing additional heat loss. This can lead to a decrease of body temperature by 0.5 °C–1.5 °C within the first 30 minutes of an operation [27, 28] and may continue attenuated for up to three to five hours [28, 29]. This perioperative heat loss is linked to prolonged length of stay as well as an increased patient mortality rate [30]. It may cause nausea [31] and lead to cardiac events like arrhythmia and heart attacks [32], coagulopathy, increased transfusion requirements [33, 34] and effective duration of anesthetics [35], a higher risk of wound infections [36] and even promote pressure ulcers [37]. Furthermore, changes in the serum K^+ concentration [37] and reduction of the subcutaneous oxygen partial pressure [27] have been observed and linked to hypothermia. However, besides the general goal of maintaining normothermia, in some medical conditions like traumatic brain injuries a reduction of body head and slowed down body functions can reduce intercranial pressure (ICP) in the patient's head which has been shown to be beneficial for patient outcome [38].

2.1.4 Lab Practices and medical workflows

2.1.4.1 Arterial and venous accesses, measurements and blood withdrawal

To obtain a reliable way of administering medications and collecting blood samples, access to large and central blood vessels needs to be established [16]. An arterial access is mainly used for obtaining arterial blood samples for analysis, whereas venous accesses are used for administering of the various required drugs. To provide enough throughput for the often various and simultaneously required infusions, venous catheters with multiple lumen (individual tubes within a single catheter) are used. Additionally, these separate channels prohibit mixing of the different infusions prior to reaching the blood vessel, which might cause undesired effects or complications like clotting or flocculation. Those multi-luminal CVC have a length of about 15 cm–30 cm [16] and are advanced within the venous blood vessel to be positioned close to the heart.
To provide access for blood withdrawals and control of blood and infusion flows through the catheters, three-way taps are used. Those valves can be arranged in different configurations to route the fluid flow between each possible combination or to block it. Due to clotting effects of blood, which may accumulate within the catheter and three-way taps, throughput may be reduced, rendering drug administering and measurements inaccurate or impossible. To prohibit such problems and maintain the access, flushing of the catheter with a sterile solution may be required infrequently when indicated.

2.1.4.1.1 Seldinger technique The so called Seldinger technique is a method for insertion of catheters into blood vessels. First, a sharp, hollow needle is inserted into the blood vessel, often guided by ultrasound. Second, a guidewire is inserted into the hollow needle, which is removed once the wire is in place. Next, the catheter to be inserted is passed over the guidewire into the blood vessel. Last, the guidewire is removed through the inserted catheter, which is then fixated in place to maintain the established access [16]. This method allows a minimal invasive establishment of an arterial or venous access, minimizing infection risk and complications.

2.1.4.1.2 Blood withdrawal process Blood withdrawals can be performed through the established venous and arterial accesses. Therefore, a small amount of blood is withdrawn with a syringe from the installed three-way tap to remove residual, non-representative blood from the catheter. Then, a small heparinized syringe or a laboratory adapter and a blood collection tube is connected and the blood sample is withdrawn. Heparinization is used to inhibit blood coagulation of the blood sample, e.g., for BGA, whereas specific blood collection tubes with additives are used for various laboratory blood analyses. After successful sampling, the catheter needs to be flushed to avoid clotting of residual blood within the catheter.

2.1.4.2 Blood analysis

Given the previously described accesses to a patient's arterial and venous system and the general method of withdrawal, collected blood samples can be used for different laboratory analyses. BGA is a first, regularly performed analysis for many important parameters. Laboratory analysis provides additional information about the initial patient state and progress

by in-depth analysis of collected blood samples with methods not covered by the fast performable BGA.

2.1.4.2.1 Blood gas analysis blood gas analysis (BGA) is an important measurement to obtain information about oxygen partial pressure (pO_2), carbon dioxide partial pressure (pCO_2), pH, dissolved electrolytes like Na^+, K^+, Ca^{2+} and blood sugar as well as hemoglobin levels. Given this broad spectrum of obtained parameters, BGA is the most ordered test for patients in intensive care. It is frequently performed to allow diagnosis and management of respiratory and metabolic problems like acid-base disturbances and assessment of sufficient gas exchange and oxygen delivery [8, 15].

2.1.5 Caregivers' tasks and workload

For maintaining the high medical standards and providing the necessary care for critically ill patients, caregivers like physicians and nurses have to provide optimal care by using the presented subset of medical devices and protocols to prevent the mentioned critical conditions. Additionally, due to the patient's condition, this care must often be performed as a time-critical task. Whereas ICUs in general have an increased number of caregivers compared to other hospital wards, where understaffing is a more severe problem [7], time management and high workload are often a relevant and burdensome factor for caregivers to achieve the therapeutic goals. Given the many devices and protocols, stress and large amounts of information in the ICU environment may lead to errors despite many efforts to improve communication and information transfer with checklists, protocols and by other means [8]. However, while guiding care and reducing errors, workload is further increased and many manual tasks like collection of blood samples, adapting rates for infusion pumps or managing therapeutic anticoagulation with often still handwritten protocols and processing steps remain.

2.1.6 Drugs and Infusions

Most drugs on the ICU or within the OR are administered by IV infusion. They are used for management of anesthesia or to provide supplementary fluids and electrolytes for therapeutic purposes or account for blood loss. Ready to use infusions are most often provided in glass bottles or flexible plastic bags that are placed bedside and administered using infusion pumps. Other pharmaceuticals and fluids, are often drawn up with a syringe from small glass bottles or ampules and administered in a defined dilution using the same means. If they don't have to be administered continuously or need to be provided as a bolus at once, injection is often performed directly by hand.

2.1.6.1 Drug overview

Concerning the scope and focus of this thesis, only a short and general overview regarding the names and potential effects of the used and mentioned drugs can be given.

Regarding more specific and reliable information about applications, potential contradictions and other important aspects, please refer to medical and pharmaceutical literature.

Artrenol (Norepinephrine) causes vasoconstrictions, narrowing of the blood vessels, therefore increasing blood pressure [39, DB00368].

Atropine is a naturally occurring belladonna alkaloid used as an anti-parasympathetic drug, increasing heart rate and counteracting parasympathomimetic drugs [39, DB00572].

CaCl (Calcium chloride) is used as a diluted infusion solution for homeostasis management and treatment of critical conditions like hypokalemia [39, DB01164]

Fentanyl is a synthetic opioid processed in the hepatic metabolism that can be administered via IV infusion. It is used for analgesia and sedation, causing increased pain tolerance while decreasing the perception of suffering. It depresses the respiratory and cough reflex, reduces heart rate (HR) and causes urinary retention [39, DB00813][8].

Furosemid (Lasix ^{TM}) is used as a diuretic, promoting the voiding of urine by inhibition of water re-absorption within the kidneys [39, DB00695].

Glucose is used in solution as an IV infusion to manage the blood glucose level, as an obligatory energy source [39, DB09341]. The different dilutions are often incorporated in the infusion solution name, therefore G20 refers to a 20% glucose solution.

Heparin (unfractionated) is used as a prophylaxis of venous thrombosis and pulmonary embolisms by inhibiting clotting of the whole blood. Therefore, it is also used for preserving blood samples for several laboratory analyses [39, DB01109].

Jonosteril^{TM} is the trade name of an infusion solution marketed by Fresenius, providing fluid and electrolytes in balanced and acidic conditions [40].

KCl (potassium chloride) is used in dilution for homeostasis and ion-concentration management [39, DB00761].

Ketamine is a general anesthetic that can be administered via IV infusion and is processed in the hepatic metabolism. It is not a skeletal muscle relaxant and may cause hallucinations and increase HR and BP among other effects [39, DB01221][8].

Midazolam is a benzodiazepine used for depression of the central nervous system (CNS) that is processed by the hepatic metabolism and can be administered via IV infusion. It is used for sedation and amnesia, acting as relaxant and anti-seizure drug [39][DB00683][8].

NaBic (Sodium bicarbonate) is used in solution as a buffering agent for treatment of metabolic acidosis, which may be caused by circulatory insufficiency, shock or severe dehydration [39, DB01390].

NaCl (Sodium chloride) is used in isotonic watery solution as a source for electrolytes and water while maintaining the intra- and extracellular osmotic balance. Additionally, it is often used as the foundation for dissolving and diluting other pharmaceutical additives [39, DB09153].

Propofol is a sedative-hypnotic agent and can be used for introduction of general anesthesia with a rapid onset of hypnosis about 40 seconds after injection [39, DB00818].

Tris (Tris(hydroxymethyl)aminomethane, THAM) [39, DB03754] is used as a buffering infusion for treatment of metabolic acidosis.

2.1.7 Common units of measurement

Within the medical context of this thesis, several parameters using different units of measurement will be discussed. To provide a basic understanding of the relevant and often non-SI units for measurements, a brief overview is given in Table 2.1.

Table 2.1: Brief overview of common units for measurements within the medical context.

Unit	Description
mmol/L, μmol/L	Molar concentrations of substances
mmHg	millimeter mercury pillar common pressures unit for fluids (e.g., blood pressures)
mbar	millibar common pressures unit for gasses (e.g., ventilator)
U, IU	International Unit (IU), used for defined pharmaceutical doses
U/kg	Unit per kilogram, used for doses related to patient weight
U/kg/h	like above, with time relation for continuous administering
1/min, bpm	beats or events per minute e.g., heart or respiratory rates
kg BW	kilogram body-weight used for describing relations to body weight in kg

2.1.8 Animal Studies and Porcine Model

Animal studies may be used and required if the understanding of a specific problem is not sufficient enough for study on humans and no other means like sufficient models for simulation can be used. In a well-defined study setting, scientific research to gain a better understanding of the problem, for example, regarding critical conditions that might be rare in human patients but essential for understanding of biochemical properties can be performed. Additionally, clinical methods and devices may need to be tested prior to or as a certification for human application.

2.1.8.1 The three Rs

In 1959, a time where animal welfare regulations did not yet exist in most countries [41], the pioneering publication by Russell and Burch [42] proposed three fundamental principles for the handling and use of animals in scientific research — Replacement, Reduction and Refinement. Whereas in short but not concluding *Replacement* aims to avoid animal studies whenever alternative models or methods are available, *Reduction* should be performed to limit the amount of animal studies to the necessary amount for providing significant results

and minimize the suffering. And, last but not least, *Refinement* should be done to improve methods and study protocols to further limit the amount and burden of animal subjects during such studies.
These principles, nowadays well known among researchers and protectors of animal rights [43], became an essential part of scientific ethics and legislation, including directives and regulations by the EU [44] and the European Laboratory Animal Science Associations (FELASA). Regarding detailed interpretation and implementations, controversial opinions exist. It is up to ethics debate if reduction of total number of used animals or relative reduction by gaining more knowledge from a single animal is a desirable goal [43]. Stated as first and most important principle by Russell and Burch, replacement should be the uppermost goal regarding animal studies, however full replacement of animals is not yet possible for all scientific aspects and stated as a long-term objective by EU directives [44].

2.1.8.2 Porcine surrogate model

Whereas traditional biomedical models, like Drosophila, zebra fish and rodents, are well suited for basic genetic studies or analyzing individual effects, they often cannot represent the complexity of a more general physiological model [45] or are not applicable to devices and procedures designed for humans. In such cases other models are required.
Swine, especially young pigs with body weights of 30 to 40 kg, are one such alternative and frequently used for medical training and research [46]. They provide many anatomical and functional similarities to humans, including aspects like physiology, immune system and disease progression, especially in metabolic and infections conditions, and are good models for the cardiovascular, urinary and digestive system as well as wound healing [45–49]. Furthermore, they are accepted as general surgical model in many areas of biomedical and "'-omics"' research [45, 50, 51]. Porcine studies allow for investigations related to potential treatments and preventions strategies for many human diseases and conditions in a rigorously controlled setting [52] as nearly every medical procedure can be performed [49]. Thus, such models are approved for preclinical evaluations regarding biocompatibility and function of implanted devices for various organs and used for training of techniques that cannot be adequately simulated using computers or anatomical models [49]. Additionally, factors like availability, cost and ease of handling play an important role regarding the choices of animal models [47]. As such factors are beneficial in swine, this has led to a significant increase in using porcine models as a replacement, e.g., for canine models, in recent decades [51].
Regarding the scope of this thesis, considering the interconnection of medical device and development of automated solutions for critical care in humans, the use of medical devices that are available and approved is favorable. Thus, the used model animal should be compatible regarding body size, measured parameters, device settings and medical procedures, covering aspects like placement of catheters and collection of blood samples. Under these conditions, a porcine model is well suited and obtained results can easily be transferred to humans. Thus, female German Landrace (DL) pigs were used in the performed studies, as DL is the most common domestic breed in Germany [53] and differences as between domestic farm breeds and miniature breeds often used as models are only related to growth rate and handling factors, not regarding anatomical differences [49].

2.1.9 Physiological References

To provide references for assessment of performance for different methods and automations presented in later chapters, a brief overview of several physiological parameters as a comparison between humans and swine is provided in Table 2.2. It can easily be observed that most human and porcine parameters have similar nominal ranges. Thus, porcine models provide a good general model for human physiology. The presented porcine references for the German Landrace (DL) breed are based mainly based on the theses of Kixmöller [53] and Nebras [54]. For the former, the measured parameters are based on 50 samples of 10 to 12-week-old pigs, whereas the latter performed a more general comparison to other published references, giving raise to similar results.

Table 2.2: Reference ranges for nominal physiological parameters for humans and surrogate porcine model at rest.

Parameter	**human**		**porcine model**		**Unit**	**Sources**
	low	high	low	high		hum.,por.
pH (art)	7.37	7.45	7.45	7.51	-	[55],[56]
pH (ven)	7.35	7.43	7.40	7.43	-	[55],[57]
pCO_2(art)	32	43	37.7	42.3	mmHg	[55],[56]
pCO_2(ven)	37	50	41.8	49.1	mmHg	[55],[57]
O_2 (art)	71	104	68	74	mmHg	[55],[56]
O_2 (ven)	36	44	18.8	39.1	mmHg	[55],[57]
K^+	3.5	5.0	4.35	6.87	mmol/L	[16],[53]
Na^+	135	147	137	149	mmol/L	[16],[53]
Ca^{2+}	2.2	2.6	2.55	3.27	mmol/L	[16],[53]
Cl^-	95	105	100.42	110.62	mmol/L	[16],[53]
Fe^{2+}	6.6	29.5	12.49	34.29	µmol/L	[55],[53]
Lactate	≤ 2		0.5	11.0	mmol/L	[16],[58]
Glucose	3.9	6.1	3.7	7.3	mmol/L	[16],[53]
$\widehat{=}$	70	100	66.7	131.5	mg/dL	[16],[53]
Haemoglobin	12.3	15.3	6.3	8.7	mmol/L	[55],[53]
BE	−2	3	1.3	5.1	mEq/L	[55],[57]
Body temp.	36.5	37.5	38.7	39.8	°C	[26],[59]
ABP_m	106	114	92.7	111.3	mmHg	[60],[56]
HR	48	98	70	120	1/min	[61],[59]
RR	12	18	17.1	22.9	1/min	[62],[56]
TV(lung)	$\approx$500		$\approx$200		mL	[15],[63]
Body Fluid	$\approx$600		$\approx$600		ml/kg BW	[64],[65]

When available, references for women and female DL swine were used.

2.2 IT background and used tools

2.2.1 Software Architecture

In outlook on Chapter 7, where the developed software framework is described, a few fundamental concepts and definitions of software architecture used within this description should briefly be introduced.

Classes and abstract classes

A class is an object definition with object-oriented programming. It can provide encapsulated functionality, can contain an internal state and defined interfaces to the outside. An abstract class is an abstract definition, providing templates or reusable methods. Other classes realize the abstract objects by implementing the defined methods and using the provided methods.

Interface

An interface is a definition for the interaction of different program parts. Different classes can provide a common interface and thus be interchanged. Using interfaces provides clear separations and defined interactions for different parts of programs.

Plugin

A plugin is an encapsulated part of a program that adds specific features to a base program. For interaction with a plugin, defined interfaces can be used.

Factory

A factory is an object for creating instances of other objects. For example, a factory can be used to create instances of objects loaded from dynamic libraries during program run-time.

Smart Pointer

A pointer is a reference to an object within the memory. The allocated space for this object needs to be freed to be available again. A smart pointer keeps track to the references linking to this object within the memory and performs automatic deletion of the object once its reference count reaches zero and it thus is no longer needed and accessible.

2.2.2 Infrastructure and Networking

Besides the terms of software architecture, for understanding the implementation of the software framework within a real hardware environment, several more specific terms regarding network infrastructure and technology need to be explained briefly.

DHCP

The Dynamic Host Configuration Protocol (DHCP) is based on a client-server model where a DHCP-server allocates IP addresses and configure client hosts by transmitting host-specific configuration information within a TCP/IP network [66].

DNS

The Domain Name System (DNS) is a system used for mapping host names to IP addresses as well as defining name spaces and domains [67–69].

NTP

The Network Time Protocol (NTP) is used to synchronize computer clocks among different clients using a single timeserver or a chain of local and remote timeservers synchronized to each other [70].

VLAN

A Virtual Local Area Network (VLAN) is a virtual network within a single physical network. Multiple VLANs can be used independently on a single physical layer, e.g., with a single Ethernet cable. To allow for such an implementation, the used network infrastructure, including switches, must support the VLAN infrastructure. Managed switches can be configured to assign individual network ports to a specific VLAN. Therefore, the client does not observe any other virtual networks except the designated one. On contrary, the raw connection, the *trunk*, can be forwarded to the client. The client's network card can then select the specific desired VLAN [71] itself.

Link Aggregation

Link Aggregation or Trunking is a method of combining multiple network connections to a single logical connection, or a Link Aggregation Group (LAG), therefore increasing bandwidth and reliability as well as sharing the load between the individual links of the LAG. It is defined by the most recent IEEE standard 802.1AX-2014, superseding previous standards for Link Aggregation [72].

Spanning Tree Protocol

The Spanning Tree Protocol (STP) is part of a smart switch infrastructure and used to prevent loop-backs in Ethernet networks. It is defined in the most recent IEEE standard 802.1D-2014. The original STP and the more recent successor Rapid Spanning Tree Protocol (RSTP) are used to provide fault tolerance through automated reconfiguration of the network by finding the fastest link to a client, therefore besides other means, preventing the creation of network loops that would cause network problems [73].

2.2.3 Qt

Qt [74] is a cross-platform software development framework for embedded and desktop, based on C++. It is available as a commercial or open-source variant. For the implementations described within this thesis the open-source variant was used by dynamically linking the executable files to the Qt libraries [74].

2.2.3.1 Signal-Slot concept

The Qt's Signal-Slot concept [75] is a fundamental feature of the communication between different objects within the Qt framework and its meta-object system. The general idea is to allow a communication between different objects as an alternative to callback functions

often used by other frameworks. Pre- or user-defined signals can be emitted by an object. Other objects can register to listen to the emitted Signals by a callback method, the Slot, which is triggered if a signal is received [75]. Therefore, the Signal-Slot system is a beneficial simplification to the general concept of subscriber patterns, where each object has to perform the management and processing of information and clients itself. It provides a good abstraction layer for interaction of different objects or plugins and is used in the developed medical software framework described within this thesis.

2.2.4 R

The R language and programming environment is managed by the R Project for Statistical Computing [76] and provides a powerful open-source software for statistical computing with a broad variety of statistical, analytical and graphical tools for processing and visualization of scientific data [77]. Besides the multitude of integrated features, the program scope can be further expanded by using the various available libraries for specific problems and calculations provided by a large active scientific community.

2.2.5 Data formats and Protocols

For storage of the accumulated data, a variety of data formats and protocols may be used. Besides commonly used and well-known formats like comma-separated values (CSV), binary or Extensible Markup Language (XML)-encoded files, more specific formats used for data gathering and processing will briefly be described in the following.

2.2.5.1 HDF5

The HDF Group [78] aims at supporting mission-critical and scientific needs for accumulation of data with open source software since 1989 with the goal of providing technologies for managing large and complex datasets [78]. The hierarchical data format (HDF), currently in the most recent version 5, provides a flexible data model for representation of complex data objects and meta-data in a file-based, portable, platform-independent format [79]. The HDF5 software library provides an application programming interface (API) for implementation in a broad variety of programming languages. Its format allows a theoretically unlimited file size for the stored data objects and provides a general, hierarchical data model with a variety of predefined data types and the ability of adding meta-data to the entries. Additionally, the HDF5 format provides a variety of compression and chunking strategies to improve the performance of data handling and storage [80]. The HDF5 file format can be processed with Matlab [81] or in R with the help of the Bioconductor `rhdf5` package [82, 83].

2.2.5.2 HL7

Health Level Seven (HL7)[84] is an international non-profit organization founded in 1987 dedicated to providing American National Standards Institute (ANSI) and International Organization for Standardization (ISO) approved standards for exchange of electronic health information and most commonly known for their standards regarding data exchange between medical devices, `HL7 Version 2.x` [85] and `HL7 Version 3` [86]. Whereas Ver-

sion 3 is based on an XML structure, the most popular and established Version 2 uses a plain-text format with the pipe or vertical bar character | as a delimiter.

2.2.5.2.1 HL7 message example A reduced example of a message from a BGA is illustrated in Figure 2.2. The message header MSH is followed by fields for the device ID, here a Radiometer ABL835 BGA device. The result is described with the ORU (observation result section), which first states a patient identification (PID), followed by a variable amount of observation results (OBX). In this example, the measurement result of the partial CO_2 level in the blood sample is reported as 41.7 mmHg.

Figure 2.2: Example for a message in the HL7 message format in the standardized version 2.

```
MSH|^~\&|ABL835^A5W - 7|ABL835^A5W - 7|||20141008103105||
ORU^R31|1|P|2.5|||AL|NE||||||
PID|1||ap003||^|||||||||||||||||||||||||||||| [...]
OBX|2|ST|^^^pCO2^M||41.7|mmHg||N|||F|||||^||| [...]
```

2.2.6 PostgreSQL Database

PostgreSQL [87] is an open-source object-relational database, based on the structured query language (SQL). A relational database is the combination of relational database schemata and stored information. The used relational schemata specify the attributes and domains of the individual relations. Information is stored by creating instances of those defined relations with the desired values [88]. Once the information is stored, specific subsets of information can be retrieved by querying the database with user-definable requests.

2.3 Analytical, mathematical & machine learning background

2.3.1 Handling missing Data

One important aspect in data processing and analysis is the handling of missing data. Especially in raw data streams, outages of individual devices or measurement errors will result in gaps within the stream. Such occurrences are often only seen as lost information and an unfilled gap [89]. Yet, despite being missing information, the absence may be caused by different reasons, allowing a classification of the type or reason.

2.3.1.1 Types of missing data

In regard to classification of missing data into different types, three distinct entities can be observed: First, data can be Not Missing At Random (NMAR). This is the case if the absence of data depends on other factors. Therefore, the absence is not at random but depending on the current state, related parameters, settings or measurements with a causal relation to the absence of the measurement [7]. Second, data can be Missing At Random (MAR). This is the case if absence of data is unrelated to the observation [7]. Third, data can be Missing Completely At Random (MCAR). Reasons for this observation may be technical failures, breakdowns that are itself completely unpredictable [7].

2.3.1.2 Data processing & imputation

For further processing of a dataset, missing data has to be dealt with. This can be performed in a variety of ways. In general, two principles are used: masking and data imputation. Whereas the first approach is used to discard the erroneous sections of the data scream, thus excluding it from further analysis, the latter one, known as data imputation, tries to calculate or predict the missing sections with the knowledge of the surrounding data in a variety of ways. Such methods include mean imputation and (stochastic) regression imputation [90]. Depending on the reason why data is missing, such methods can introduce a bias to further analysis and only if data is MCAR, data imputation results in unbiased estimates [7, 89]. Yet, another view of the problem is to see the missing observations as its own measurement outcome. Using this approach, such data points are not discarded, but further processed as its own type of information [89]. This is especially useful, if data is NMAR, thus having an underlying cause or reason and knowledge of it may be beneficial for further analysis.

2.3.2 Fuzzy Logic and Fuzzy Controllers

Fuzzy Logic (FL) is a mathematical logic first described and refined for various applications by L. A. Zadeh [91–95]. On contrary to the well-known binary logic with precise reasoning, one of the strengths of FL is being more approximate using precise definitions only as the edge cases [93]. Thus, it is well suited for relationships that cannot easily be modeled, including vague human assessments [96], and successfully used for process control since the 1980s [93]. One of the fundamental concepts of FL is establishing a relationship to linguistic variables for description of states and implications [96].

2.3.2.1 Fuzzy Sets

The foundation of fuzzy logic is the definition of fuzzy sets that can be seen as an extension of the well-known ordinary (or crisp) set theory. Like those crisp sets S, a fuzzy set F is a collection of objects within a domain $X = \{x\}$. On contrary to crisp sets, the membership is not a simple binary decision in fuzzy sets, but defined by a membership function μ, assigning a grade of membership to the individual elements x in the fuzzy set f. This graduate membership dissociates fuzzy logic from other multi-variate logic systems with more precisely and limited grades of memberships [91].

2.3.2.1.1 Fuzzy membership function A membership function $\mu_F(x) \rightarrow [0,1]$ is used to assign a grade of membership of an element $x \in X$ in a fuzzy set F within unity. This general function can be seen as an expansion of the membership in crisp sets, where only the binary mapping $f_S(x) \rightarrow \{0,1\}$ is allowed. The membership function $\mu_F(x)$ itself is completely non-statistical but has similarities to a probability or density function and often is derived from empirical observations to assign grades of membership based on statistical analysis [91].

2.3.2.1.2 Fuzzy set operations Given the fundamental definitions of the sets and membership functions for assigning grades of memberships, operations on some single or multiple fuzzy sets need to be defined. Similar to operations on crisp sets, mathematical rules for dealing with fuzzy sets can be defined as an expansion of those crisp operators.
The most fundamental set operation is the complement $F' : \mu_{F'} = 1 - \mu_F$. For the edge cases 0 and 1, this more general definition for fuzzy sets yields the same results as for crisp sets. The next fundamental operation is the union $\cup$ of two fuzzy sets A, B: $C = A \cup B$. In crisp sets, an element x is included within C if it is included in A or B. Expansion to partial memberships for fuzzy sets, is performed by retaining the higher degree of membership. Therefore, the fuzzy union $\cup$ can be defined as a membership function $\mu_c(x) = max(\mu_a(x), \mu_b(x))$ or in short $\mu_c = \mu_a \vee \mu_b$. Equality for crisp edge cases can easily be observed again. Last but not least, the intersection $\cup$ of two fuzzy sets A, B: $C = A \cup B$ is defined in a similar way, maintaining the lowest degree of membership: $\mu_c(x) = min(\mu_a(x), \mu_b(x))$ or $\mu_c = \mu_a \wedge \mu_b$ [91, 92].
Using these fundamental definitions of set operations, other properties and other mathematical laws, like distributive and De Morgan, can be defined similar to crisp sets. This provides a well-defined mathematical foundation of fuzzy set theory. Besides the presented and original definitions of fuzzy set operations by Zadeh, other operations for union and intersection can be and have been defined for specific purposes over the decades but those shall not be further elaborated upon.

2.3.2.1.3 Fuzzy functions Fuzzy functions $f = \{\mu_1, \ldots, \mu_n\}$ are a combination of different membership functions μ allowing the modeling of different grades of membership for different conditions of a single crisp variable. Often, membership functions within a fuzzy function are related to and derived from linguistic terms and expert knowledge, for example describing an observation in terms of severity with membership functions like *slightly low*, *normal*, *high* or *critical.*

2.3.2.2 Fuzzy controllers

Fuzzy controllers are controllers based on fuzzy set theory using fuzzy sets f and a rule base with IF ... THEN rules to perform control. Application of fuzzy logic in a control algorithm was first described by Mamdani [97] for a steam engine controller by extension of fuzzy logic as proposed by Zadeh with the rule base and an implication rule similar to classical binary logic. This Mamdani implication is defined as $\mu_{A\to B} = min(\mu_A(x), \mu_B(x))$ or $A \to B = A \wedge B$, limiting the implication to the minimal degree of membership for the premisses.
In general, such a fuzzy controller needs to perform three separate steps: The first step is the fuzzification of the crisp input values, mapping them into the fuzzy space. The second step is the inference, evaluating the defined rules and implications to obtain membership grades for the output functions. Finally, in a last step, the fuzzy degrees of membership of the output variables need to be defuzzified to obtain crisp output parameters. In the sequel, these three steps will be elaborated in more detail.

Guiding this general description, and example is provided and illustrated in Figure 2.3: Let's assume a vital parameter can be *low*, *normal* or *high*, based on medical knowledge. Therapy should be led by this observation and *some* medication should be administered if the vital parameter is *low* or *normal*, otherwise *none*.

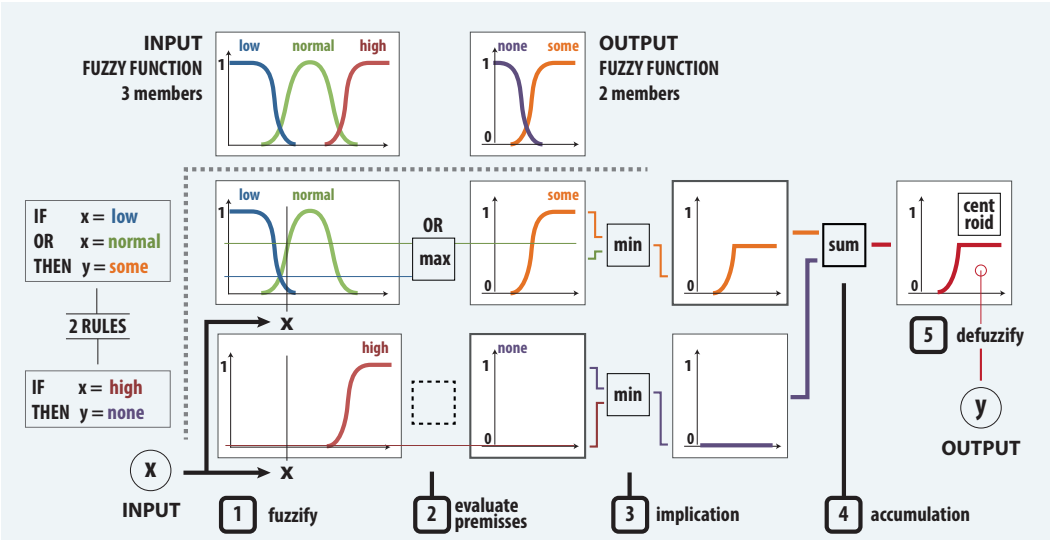


Figure 2.3: Illustration of the processing within a FL controller for a simple example using a single input for an observed vital parameter. Administering of a medication is decided by the output variable. A simple rule base with two simple rules is used and processing is performed in five steps: fuzzification (1), processing (2–4) and defuzzification (5).

2.3.2.2.1 Fuzzification The first processing step within a fuzzy controller is transforming the crisp, numerical input values x_i to membership grades of respective fuzzy functions f_i. These fuzzy functions f_i are defined for any input variable x_i and often incorporate expert

knowledge. The member functions μ_{fi} within a fuzzy function f_i are often equally spaced triangular functions as simplifications of Gaussian curves. At the boundaries of the input range, trapezoid functions as approximations of sigmoid curves are often used. By evaluation of each input value x_i within its respective fuzzy function f_i, the fuzzy membership grades can be obtained $f(x_i) = (\mu_{fi,1}, \ldots, \mu_{fi,j} \ldots, \mu_{fi,n})$. For the assumed working example with the measured vital parameter, three membership grades $\mu_{low}, \mu_{normal}, \mu_{high}$ would be calculated for a present input value.

2.3.2.2.2 Rule application and implication Given the membership grades $f(x_i)$ of each input variable x_i within its corresponding fuzzy function f_i and its membership functions, evaluation of the input can be performed. The rule base consists of a set of simple *IF–THEN* rules, where conditions and implications refer to the individual membership functions. Evaluation of all rules is performed in parallel in two steps: In a first step, premisses of each rule are evaluated. They consist of a single membership function of an input variable or a combination of individual variables of multiple inputs. Interference of those variables can be performed by logical operators like AND or any other well-defined operator within fuzzy logic. The result of the evaluation of the premisses for a single rule is a grade of membership $m = \mu(x)$. In the second step, rule implication using the defined implication operator is performed. For Mamdani controllers, this implies limiting the grade of membership to the minimum of premises membership and the output membership function. In the working example the rule base would consist of the two rules for implication:

$$
\begin{array}{lllll}
1: & \text{IF } x = low & \text{OR } x = normal & \text{THEN } y = some \\
2: & \text{IF } x = high & & \text{THEN } y = none
\end{array}
$$

The measured vital value x would be evaluated regarding those two rules to: $m_{low \wedge normal} = min(\mu_{low}(x), \mu_{normal}(x))$ and $m_{high} = \mu_{high}$. Implications would be performed for both rules, resulting in two membership functions $\mu_{y1} = max(m_{low \vee normal}), \mu_{some}$ and $\mu_{y2} = min(m_{high}, \mu_{none})$.

2.3.2.2.3 Defuzzification The last step within a fuzzy controller is defuzzification to obtain a crisp output. The parallel rule application and implication results in respective membership functions, which need to be combined to obtain a single output. This step, accumulation, can be performed with well-defined fuzzy operators like multiplication or sum, resulting in a single membership function. Now, only conversion of a membership function to a crisp value remains to be done. This step, called aggregation, is performed with fuzzy operators like max, min or centroid. The defuzzified result then is the output of the fuzzy controller. If more input variables and multiple outputs are used, this process is performed in parallel for each output variable. In the used example, accumulation would combine the two membership functions μ_{y1} and μ_{y1}, e.g., using a maximum operator $\mu_o ut = max(\mu_{y1}, \mu_{y2})$. Aggregation then could be performed by finding the centroid of this membership function providing the output value $y = centroid(\mu_o ut)$.

2.3.3 Linear regression

Regression aims to find a function $f(x) = \beta \cdot x + b$ to predict an output $f(x) = Y$ by calculating a response from an input vector X, using regression coefficients β and an intercept b. A well established and one of the most important regression methods in statistics for almost 40 years is linear regression [98]. Normally, a least squares approach is used to determine regression coefficients β, minimizing the residual sum of squares using the Euclidean or L2 norm $||\cdot||_2$. While being a simple approach, adequate and interpretable results can often be acquired and even outperform more complex models with sparse or limited training data [98]. The general form of linear regression for finding optimal coefficients $\hat{\beta}$ for prediction of a response vector $y \in \mathbb{R}^n$ using an input vector $X \in \mathbb{R}^p$ can be described as follows:

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} = \|y - X\beta\|_2^2$$

2.3.3.1 Ridge regression

Ridge regression aims to counter problems like large variances as a tradeoff for providing a low bias present in common linear regression by regularization. Regularization tries to minimize the coefficients by imposing a penalty on the size of the β coefficients, a concept similar to weight decay in neural networks [98]. Thus, ridge regression is also known as Tikhonov regularization. Similar to simple linear regression, ridge regression uses the Euclidean L2 norm for minimization and utilizes a complexity parameter λ to control the strength of regularization. For $\lambda = 0$, the penalty term is eliminated, resulting in common linear regression. If an intercept term is included in the regression model, this term is normally excluded from the regularization penalty. Due to the imposed penalty term $\|\beta\|_2^2$, an optimization bias for different numerical ranges of the regression coefficients can occur and should be avoided for larger coefficients by standardizing the inputs [98]. Due to the used L2 norm $||\cdot||_2$, no variable selection can be performed as ridge regression can only shrink coefficients towards zero but not eliminate them entirely. Like for common linear regression optimal coefficients $\hat{\beta}$ of the ridge regression using the L2 penalty term $\lambda\|\beta\|_2^2$ are determined by:

$$\hat{\beta}_{ridge} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} = \|y - X\beta\|_2^2 \; + \; \lambda\|\beta\|_2^2$$

2.3.3.2 Lasso regression

Lasso (Least Absolute Selection and Shrinkage Operator) regression is another deviation from common linear regression first described by Tibshirani [99]. Whereas ridge regression is able to perform regularization by introducing a penalty term using the L2 norm $||\cdot||_2$, it is not able to solve the problem of variable selection. Lasso solves this problem by a small modification of the ridge regression through regularization using the L1 norm $||\cdot||_1$, also known as Manhattan distance. Using this tweak, regression coefficients can be set exactly to zero. This can significantly aid interpretation in problems with a large number of predictors [98]. Similar to ridge regression, Lasso regression solves the optimization problem using the L1 norm in the penalty term $\lambda\|\beta\|_1$:

$$\hat{\beta}_{lasso} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} = \|y - X\beta\|_2^2 \; + \; \lambda \|\beta\|_1$$

2.3.3.3 Elastic-net regression

Another variation of the linear regression methods described beforehand is elastic-net regression, first described by Zou and Hastie [100]. As both ridge and Lasso regression provide benefits and drawbacks under different conditions, elastic-net regression aims at finding an optimal solution combining both approaches with an additional weigh factor $\alpha \in [0, 1]$. For the extrema of α, elastic-net regression is equal to ridge regression ($\alpha = 0$) and lasso regression ($\alpha = 1$) [98]. Combinations of the two regression types can be described by adaption of the weighting factor α. With $\lambda_1 = \alpha \cdot \lambda$ and $\lambda_2 = \frac{(1-\alpha)\cdot\lambda}{2}$, the optimization problem can then be described as follows:

$$\hat{\beta}_{elastic\ net} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} = \|y - X\beta\|_2^2 \; + \; \lambda_2 \|\beta\|_2^2 \; + \; \lambda_1 \|\beta\|_1$$

2.3.3.4 Robust regression

Models based on least-square optimization are sensitive to outliers, which may be caused by artifacts and disturbances uncorrelated to the observation. To counter those effects, more robust regression methods have been developed. One approach is using bi-squared weights, as it includes the commonly used least-square fitting while simultaneously providing a more robust regression by minimization of outlier effects. To achieve this, a weight w_i is included in the calculation of the sum of square weights for each observation. The optimal solution can be found by an iterative re-weighted least squares approach, in which a weighted least square regression is iteratively performed with updated residuals and weights until a convergence is reached. Using bi-square weighting, observations close to the fitting line will get small error weights, whereas points further away are associated with reduced weights [101, 102].

2.3.4 Support Vector Regression

Support Vector Regression (SVR) is a nonparametric approach that can be used to find an estimator function. A common way to obtain a solution is by using linear epsilon-insensitive SVM (ε-SVM). In this method, optimization is performed to obtain a solution where for all trained points none deviates more than ε from the known target and deviations within the ε range are not penalized [103]. Therefore, similar to SV classification, only a subset of training points is considered for defining the solution [104]. Additionally, a regularization is performed by solving the minimization problem of finding a function that is as flat as possible [104]. Thus, SVR aims to achieve a minimal generalization error bound (a combination of the training error and a regularization term) instead of minimizing the total training error itself [104]. Given an input vector $X \in \mathbb{R}^n$, a response vector $y \in \mathbb{R}^n$ and an

intercept b, the ε-SVR minimization problem can be described as follows:

$$\begin{aligned} & f(x) = \beta x + b \\ \text{minimize}: \quad & \frac{1}{2}\|\beta\|_2^2 \\ \text{subject to}: \quad & \forall i : |y_i - (\beta \cdot x_i + b)| \leq \varepsilon \end{aligned}$$

For non-linear regression, the problem can be solved by mapping the input space to a higher dimensional feature space or using an optimized solution utilizing kernel functions in the feature space without the need of mapping to the higher dimension itself (kernel trick). Unfortunately, when solving a SVR problem, a solution where the ε margins can be fulfilled for all points may not exist. To obtain a solution for those cases, softer margins allowing penalized error margins larger than ε can be introduced by using so slack variables ξ [103].
Given the description of the minimization problem, the final step is algorithmic solving. This can be performed in various ways, sequential minimal optimization (SMO) [105] being the most popular inter alia due to its good general convergence properties [103]. It splits the problem space in chunks of size to and tries to optimize the target function with respect to those sets [103].

2.3.5 Statistics and Performance metrics

Given a trained model with calculated predictions or an implemented closed-loop automation, assessment of the performance is a final crucial step. For this task, a broad variety of methods and performance metrics can be used. The ones relevant to the results of this thesis will briefly be introduced in the following.

Mean absolute deviation

The mean absolute deviation (MAD) is a relative measure for deviation from the mean. For an observation $X = (x_1, \ldots, x_n)$ with mean $\bar{x}$: $MAD = \frac{1}{n} \sum_{i=1}^{n} |x_i - \bar{x}|$

Interquartile range

The interquartile range (IQR) is a statistical dispersion measure. It is a metric for the difference between lower and upper quartiles Q: $IQR = Q_3 - Q_1$.

Standard error of the mean

The standard error of the mean (SEM) is a measure for the dispersion of individual observations around a population mean. It is calculated from the standard deviation s divided by the squared root of the population size n: $SEM = \frac{s}{\sqrt{n}}$.

(Root) mean square error and relative deviation

The mean square error (MSE) is a quality measure for an estimator $\hat{y}_i$ by calculating the mean average of the squared deviations from a desired target vector y: $MSE =$

$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$. The root mean squared error (RMSE) $RMSE = \sqrt{MSE}$ applies an additional extraction of the squared root of the MSE.
The two previously described measures are measures for the absolute deviations and errors. For a relative description of the error, the normalized root mean square deviation (NRMSD) can be used by dividing the $RMSE$ by the range of the observations of y: $NRMSD = \frac{RMSE}{y_{max} - y_{min}}$.

Coefficient of determination

The coefficient of determination r^2 is the squared correlation coefficient r between two variables and provides a measure for the strength of linear relationship between those.

Sensitivity, specificity, F1 score

Given a classification with a known ground truth, the performance of the model can be evaluated regarding the correct classifications if several or all class memberships are known. In general, four different classification results can be distinguished: true positives (TP), false positives (TN), false negatives (FN) and true negatives (TN). Based on a subset or all of those 4 groups several metrics can be calculated, those include:
The sensitivity or true positive rate (TPR) as a measure of fraction of correct hits among all events of the ground truth, precision or positive predictive value (PPV) yielding the fraction of correct hits among all detected hits and accuracy (ACC) as a measure of the fraction of true predictions among all predictions. As a deviated measure, the F1 score combines sensitivity and precision using the harmonic mean.

$$TPR = \frac{TP}{TP + FN}$$

$$PPV = \frac{TP}{TP + FP}$$

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}$$

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

3 State of the art and related work

3.1 Existing data sources

Analysis and predictions are always only as good as the potential within the underlying data sets. Therefore, a large data base and the availability of training data sets, especially for machine learning approaches, is beneficial. Unfortunately, publicly available collections of medical data are very limited due to difficulties in data collection and patient's privacy rights. One of the few available data sources is the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC) database in its second (MIMICII) or third (MIMICIII) version, providing a broad variety of patient information and vital data [106–108]. However, available data is often oriented towards diagnosis and patient identification. The provided vital data is most often limited to few parameters with measurements once an hour or limited to uncorrelated high-resolution sequences of individual parameters. Additionally, as the protection of patient rights and anonymity is of highest priority, (meta-) analysis methods that might be able to correlate different datasets and match patient data, yielding additional and more complete information of data sets for individual patients, are limited or prohibited.

3.2 Data collection

Given this lack of data, the need for collecting and accessing the various medical data sources has emerged over the last decade to facilitate event prediction and decision making for improved and personalized care [7]. One problem for data collection within the medical field is the lack of standardization and interoperability among medical devices [7, 10, 11]. Observing the technical evolution of such clinical systems, they mostly have been independently designed to document clinical activity for liability and billing reasons [7]. This yielded a heterogeneous landscape of record systems and databases, providing a variety of data sources including laboratory results, real-time monitoring devices, image scans, free text nodes, waveforms, billing codes, procedure codes and other formats [7]. Unfortunately, whereas efforts for data integration were successfully performed in other industrial fields [9], the heterogeneous landscape in the medical field continues to exist. Fundamental data integration for documentation and billing is only achieved with higher-order system handling and converting the various obtained data formats as well as manual input. As the heterogeneous data sources might change at any time, those systems need to be adapted dynamically and are often still limited [7]. Generating a more homogenous data landscape would require well-defined standards for data transfer between different devices of various cooperative manufacturers [7, 11]. Additionally, devices like the ones used for POC tests, often only provide retrospective bulk data export and have no capabilities of directly

accessing the results like those of ACT measurements.
Given those complications, collection of data mostly has been performed by researchers only for their specific needs. Yet, several more general concepts for data collection have been proposed. Norris et al. [109] suggested a network-based system for collection of alarms from individual devices and notification of caregivers. Pölsen et al. [110] presented a concept for a web-service based plug-and-play system for medical devices with a decentralized structure. Another system, based on individual laptops connected by WLAN to a central database, was developed by Feng et al. [111]. Further applications with the goal of data collection include telemedicine [112] and management of clinical workflows [113].
Regarding the heterogeneous landscape and incompatibility of medical devices, approaches to unify this landscape gain increasing attention within the recent years as an enhanced focus in the computer science world is put on the medical field and new IEEE standards like IEEE 11073 [114] for communication between medical devices are under development.
Looking beyond the box of medical research, a prominent example for an automation framework in the context of robotics is the robot operating system (ROS) [115], providing a client server-based system with message-based communication. Whereas, general application to other fields like the medical environment may be possible, many fundamental aspects of the system, including sensor-physics, path-planning and visualizations, are specifically tailored for its intended application and would provide additional overhead. Additionally, ROS is developed for Linux clients, whereas many medical devices run within a Windows environment where drivers and libraries for interaction with the medical devices often only are provided for Windows operating systems by the manufacturers.

3.3 Data processing, filtering and decision-making

Once clinical data is made available one way or the other, further challenges arise. Collecting information from different databases, which itself is a challenging task, provides no guaranty to obtain usable data sets as information from the different sources often cannot be linked with the other information [7]. This is frequently caused by the lack of common identifiers or time offsets due to the heterogeneous data sources that hardly can be corrected retrospectively. Yet, if a comprehensive set of data can be obtained, another subsequent challenge is data corruption. Clinical settings and the operation room are a hostile environment for signal acquisition and processing [116]. Whereas in other domains data is often collected by researchers for analysis and machine learning, clinical data is intended to facilitate patient care and not data analysis. Raw data collected from the multitude of sensors and data sources most often contains artifacts from sensor drop-offs, treatment interventions and incomplete measurements. Paired with the often fuzzy nature of medical data this provides additional challenges for secondary data use and analysis [7].

To deal with artifacts, a broad variety of approaches for filtering and sensor fusion have been developed [117]. This is an important step, as pre-processing can even be more important than the use of nonlinear machine learning classifiers capable of capturing higher order interactions during further analysis [118]. Often median filters [119, 120]

are used as an initial filtering approach and for reduction of false alarms. Furthermore, Kalman filters can be used for sensor fusion and filtering for short term errors and drifts [121–123] or be included in methods for event detection [124]. Other approaches utilize fuzzy-classification [125], regression-based templates [126], robust regression [127], auto-regressive models [128, 129], phase-space embedding [130] or transformations into the frequency domain [131]. For data sources with a high temporal resolution, like pressure curves, wavelet analysis [132, 133] and localized fittings [134] approaches can be applied. This can be used for artifact detection and filtering, e.g., removal of blood-sampling events from the monitoring data stream [135].

Another and often subsequent task is the closed-loop control of individual parameters based on pre-processed measurements. Within the intensive care environment, a broad variety of specific approaches were developed during the recent decades. This research includes the development and evaluation of closed-loop systems for decision support in anesthesia, glucose management, ventilation and fluid management [136].

Various successfully evaluated approaches for medical decision making are based on fuzzy logic. Examples for such applications are fluid-administering [137], assessment for urgency of care, e.g., based on mean ABP and oxygen saturation levels [138]. Even more complex models use fuzzy decision trees and cognitive maps for dosage of radiation treatment [139] or an adaptive neuro-fuzzy inference system (ANFIS) to train a set of fuzzy rules for detection of hypovolemia [140].

Additionally, a few more specific aspects of care, covered by more comprehensive research regarding management and closed-loop control, are now presented in more detail.

3.3.1 Temperature management

Targeted temperature management (TTM) can be performed by different means, including different types of surface cooling systems or intravascular systems using special catheters placed in the venous bloodstream [141]. Such invasive systems mostly include temperature sensors and perform closed-loop control to maintain the desired temperature [141]. Furthermore, systems used during dialysis can perform warming of the blood to a desired level. Non-invasive systems can achieve heating or cooling by utilization of tempered water or air. In water-based systems, water is circulated through fine tubes in special blankets by a control unit including a water pump and a heating/cooling unit. Those specialized systems are mainly used for patient cooling and rewarming for prevention or reduction of the effects of brain injuries by reduction of ICP. In general, they can be equipped with temperature sensors for closed-loop TTM and allow for a fine adjustment of the desired temperature.

Another type of systems are forced-air warming devices with special air-filled blankets providing a continuous stream of air with a defined temperature to the patient's skin [28] to allow for a defined warming or cooling and prevention of conductive and convective heat loss [142]. Those systems are smaller and more portable than the water-bases ones and only require inexpensive disposable blankets, allowing cost-effective and more widespread application in different hospital settings. However, those systems only provide limited temperature settings and no closed-loop control.

Regarding the stabilizing of study conditions in animal studies, temperature management was successfully applied by various researchers [143–145].

Besides special temperature management devices, an example for interconnection of medical devices for temperature control is the research by Britto et al. [146]. Using data obtained from patient monitoring, infusions were automatically stopped if body temperature of infants fell below a defined threshold [146].

3.3.2 Glucose management

Control of glucose levels and administering of insulin is one of the most developed fields regarding closed-loop control in the medical sector. Yet, there is still no superior method or designated algorithm for treatment of hyperglycemia or reduction of glycemic variation [147]. Several researchers have evaluated different filtering methods to improve glucose monitoring [121, 148] as the first essential step for glucose management. For this management, beside simple approach based on the manual glucose control using look-up tables, several methods using proportional–integral–derivative (PID) controllers or model-predictive control have been proposed [147, 149]. One simple example is the open-loop *ePtotocol-insulin* algorithm by Thompson et al. [150] that can be run on a dedicated laptop computer. Caregivers can input the currently measured glucose level and a new infusion rate is automatically calculated by the algorithm based on the recent input and historic values. The obtained new rate can then manually be set at the infusion pump. The algorithm uses a classification approach for the observed measurements and calculates a response based on the rate of change. Using this approach, severe hypoglycemic events were only observed in 1.42 percent of the 12.886 performed glucose measurements and adaptions based on the algorithmic suggestions [150]. Other proposed solutions use PID or model-predictive controllers. Such algorithms require little patient-specific information and are very adaptive [147]. Plank, Pachler et al. [151–153] were able to develop a model predictive control (MPC) algorithm for tight glycaemic control (TGC). Evaluation of the proposed solution was performed in clinical trials with 50 patients for about 72 hours each using an average sampling interval of 117 minutes. Control was performed by a laptop-based solution connected to the infusion pumps with manual data input. The algorithmic predictions and adaptions were based on the measured glucose levels, administered insulin infusions and carbohydrate content of parenteral and enteral nutrition [153]. Regarding the performance of the proposed algorithm, only a single hypoglycemia event with a measured glucose level below 40 mg/dL occurred in the automatically controlled group, whereas with manual control, no such event was present [152]. Further clinical evaluation with direct algorithmic integration into infusion pumps system was performed by Kulnik et al. [154]. Their evaluation included 10 patients with a median therapy duration of 66 hours. Nominal glucose levels between 80–110 mg/dL could be achieved in 47 percent of the observation time. Based on those studies, the commercial B. Braun *Space GlucoseControl* [155] is marketed. Other commercially available open-loop systems based on PID control include *EndoTool* (Monarch Medical), *GlucoCare*, *Glucommander* (Glytec, LLC) [156], *GlucoStabilizer* (Medical Decision Network, LLC) [147].

However, common to all automated approaches is the requirement for frequent glucose

measurements, often exceeding 18 measurements per day [147]. Additionally, those solutions often cause additional workload due to interaction with the devices and manual parameter input [147]. Thus, insulin and glucose management is still an ongoing research field, requiring substantial expert input, training, and supervision [149]. Hence, solving the problem of automated data collection and processing could reduce manual interaction and workload [147]. Whereas the already presented solutions are used in a clinical setting and mainly dependent on manually obtained samples for glucose measurement, another field is the long-term application for diabetes management. In such applications, minimal-invasive, implanted sensors under development or already marketed [148] can be used for continuous glucose monitoring (CGM) within the research context of closed-loop artificial pancreases [157].

3.3.3 Anesthesia and blood pressure management

Another well-established field with a strong research focus is computer-aided management of anesthesia. As management of depth of anesthesia has been a well-defined goal for a long time [116], a broad variety of automated approaches have been developed and evaluated during the recent decades [12, 116, 158–170].
One of the biggest challenges in anesthesia management are difficulties in data access and control of the devices. Paired with the vital role of anesthesia management, additional limitations regarding the scope of functions and possible adaptions have to be considered [116]. With technological advantages, ventilation devices gained more and more internal closed-loop capabilities for maintaining a defined state and experimental approaches using rule-based systems and neural networks have been evaluated. Nowadays, management of anesthesia depth is mostly based on the bispectral index (BIS) as a measure for brain activity and consciousness with closed-loop control to minimization BI variability [171].
Allen et al. [116] developed a neuro-fuzzy closed-loop system for control of depth of anesthesia that provides a powerful tool for incorporating practical experience of physicians. Their goal was to achieve a desired anesthetic depth as fast as possible and maintain this level regarding changing conditions during surgical stimulation with auditory-evoked potentials (AEP) from EEG signals. To achieve the control, a simple three-layer perceptron network with subsequent fuzzy controller, processing the outputs of the neural network, was used [116]. Hemmerling et al. [165] developed an automated closed-loop solution for the administering of intravenous anesthesia to maintain a target BIS and with their *McSleepy* system based on a single PC with touchscreen show an improved performance in comparison to manually administered anesthesia. Their data collection was carried out by a measurement every 5 seconds and writing the obtained data into a spreadsheet. Furthermore, management of ABP has successfully been evaluated by Luginbuhl et al. [159], controlling the depth of anesthesia.

3.3.4 Respiratory management

Automated and adaptive systems for mechanical ventilation provide a potential solution for problems like long ventilation times in an aging population paired with the problem of understaffing and increased healthcare cost [172, 173]. In the context of faster weaning from mechanical ventilation, protection of the lung, while still providing sufficient ventilation, is one of the most important aspects. To achieve this goal, systems utilizing a dynamic combination of different ventilation modes with automatic and frequent adaptions of the settings are used [163].
Nemoto et al. [174] developed a decision support system using a weaning algorithm based on fuzzy logic, which included measurements from heart rate, arterial oxygen saturation, TV and RR. All input parameters were fuzzified and the current state and optimal action to be applied was determined by an extensive rule-base obtained from empirical knowledge.
Therani et al. [175–177] developed and improved an open-loop decision support system for management of arterial blood gases to aid care and reduce weaning time. Data acquisition for control is limited to manual input of the required parameters by the physician. Using a rule-based system and the breathing model by Otis et al. [178], optimal ventilation levels were then calculated based on the input and suggested to the caregiver for application.
Commercially available systems using dynamic ventilation patterns and closed-loops are *VentAssist* (Philips), *SmartCare* (Dräger), *INTELLiVENT* (Hamilton Medical): *VentAssist* [179] uses a fuzzy logic inference system for evaluation of respiratory parameters, including a trained neural network to estimate the patient's breathing effort [180]. *SmartCare* is able to apply a weaning protocol in pressure support mode by adapting pressure to allow for spontaneous breathing efforts. Therefore, average RR, lung volume and end-tidal CO_2 level are evaluated every 2 to 5 minutes and the next action is suggested to the caregiver [181]. *INTELLiVENT* is able to perform continuous adjustment of mechanical ventilation to require the least work of breathing based on knowledge obtained from various clinical trials [181]. Using this dynamic adaption, better performance than classical, static modes has been observed in various studies [173, 181, 182].

3.3.5 Predictions of blood gas parameters

Several studies covering the prediction of arterial blood gases from venous samples were conducted [183, 184]. Using regression methods, high overall correlations ($r > 0.9$) could be achieved in various studies for pH, pCO_2, HCO_3. No significant correlation between arterial and venous pO_2 and O_2 saturation measurements was observed. Yet, besides such comparisons between arterial and venous blood samples, obtained from central veins or earlobes, no direct comparison of the various arterial blood gases to possible surrogate parameters, obtained as high frequency data from different patient monitoring devices, can be found in scientific literature.

3.4 Preliminary work at the Schenk Lab

Given a long-standing experience regarding porcine models in the Schenk Lab at the Institute of Experimental Surgery at the Department of General, Visceral and Transplant Surgery at the University Hospital Tübingen (UKT) [185–190], ideas and concepts for standardization and automation of clinical protocols and the need for data collection had already been considered. First tests of closed-loop controls and fuzzy logic applications for automation of different parameters were performed on distributed systems with individual laptop computers. Thus, problems like data compartmentalization and time synchronization still remained in those initial attempts for data collection from medical devices, central data presentation and closed-loop control using PID controllers and fuzzy logic. An example for the exploratory approach using a graphical pipeline within DASYlab, a graphical laboratory data acquisition and control system, is shown in Figure 3.1. This research provided the technical foundation as well as a thematic guidance for the scope and solutions presented within this thesis.

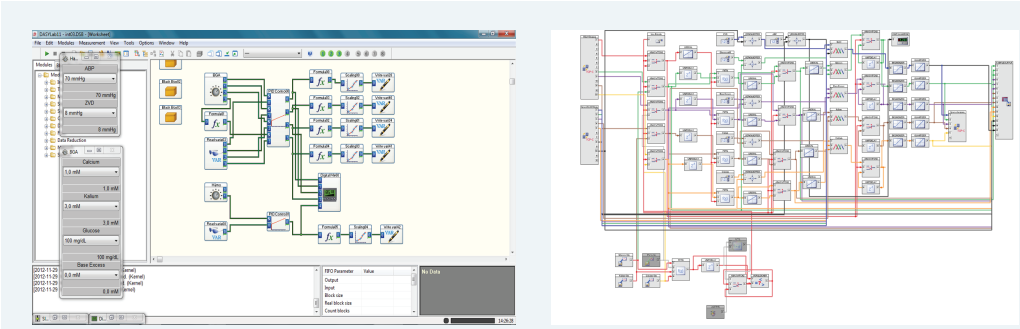


Figure 3.1: Illustration examples to provide an impression of the performed initial exploratory tests of closed-loop control using block-based graphical data flows in DASYlab with various connected modules for fuzzy-logic and PID controls.

3.5 Summary of the State of the Art

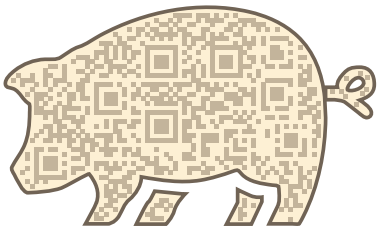
Availability of clinical datasets is limited and often restricted to small sets of collected parameters and low data collection rates. Additionally, such datasets are often oriented towards clinical diagnosis and as a measure to protect patients' rights, further analysis and correlation of different datasets to obtain additional or correlated information is limited or prohibited.

Whereas standardization is well established in many other areas of research and industry, a heterogeneous landscape continues to exist due to predominantly independent developments in the medical field [7, 10, 11]. Due to this lack of interoperability among medical devices, collection of clinical data for further processing and research is a difficult task. To close the data gap, different concepts for interconnection of medical devices have been proposed, including network-based alarm systems [109], web-service based data-collection [110] or individual laptops connected to a central database by WLAN [111]. Additionally, new IEEE

standards like IEEE 11073 [114] to improve device interconnection are under development. Overall, those systems are designed for data collection and not intended for the additional task of controlling and configuring medical devices as it is required for closed-loop applications.
Once data is collected, further processing needs to be performed as the raw data stream contains numerous measurement artifacts and other disruptions. Therefore, various filtering and sensor fusion approaches have been developed and optimized for specific parameters. Regarding the measurement frequency, blood samples generally pose a major problem as this procedure is a manual and invasive task. Yet, for the specific problem of analyzing BGA results, only correlations between various arterial and venous blood samples but no comparison to real time data of various other parameters from patient monitoring are described in published literature.
Beside data collection and (retrospective) analysis, the collected information can be used for decision support and therapy guidance. For modeling of physiological parameters and relations, fuzzy logic is an often used and successfully applied approach. Open- and closed-loop approaches for intensive care have been developed for different specific parameters like temperature management, glucose or insulin management, control of depth of anesthesia or respiratory management. Yet, all those solutions are specifically tailored and limited to individual parameters and not integrated into a general software framework or designed for the interaction with other systems. Instead, and of course by accounting for the legal limitations for clinical application on patients, those systems are standalone solutions, still requiring manual data input and adaption of the target devices by the caregivers.

Part II

Conducted Studies



4 Overview

The assessment of goal-directed closed-loop management in intensive care medicine by developing a data integration and processing framework and retrospective analysis was performed at the Institute of Experimental Surgery at the University Hospital Tübingen (UKT) alongside and within the context of ongoing clinical research using a porcine model. This setting provided the available and legal framework for testing the developed software and, as swine have a similar physiology and size as humans, allowed the usage of medical devices intended for human application.
The two clinical studies associated with this thesis were not specially designed or conducted for the task of automation but guided by different clinical research questions like the evaluation of medical products or the assessment of patients' volume states. Thus, these studies are presented as background information which will be referenced to in subsequent chapters dealing with the development and evaluation of the proposed software and closed-loop applications within those framing conditions.
Within this given study context, the goal was to establish a system for collecting all available data and use automation to stabilize and standardize study conditions by management of hemo- and homeostasis within and between the individual animal subjects to obtain more reliable research data and thus reduce variance. Therefore, reducing the need for an increased number of animal subjects. As those clinical questions and research topics guided the technical implementations and provided a focus for the desired solutions, the progress of those studies and software development of course often advanced hand in hand. Furthermore, many practical aspects relevant to those conducted studies, like repeated manual tasks according to protocols, were evaluated and subject to automation to limit the influence of human variance and error potential and reduce caregivers' workload.
In the following, a short overview of the two relevant studies will be given. This provides the general context guiding the progress of this thesis and frames the used medical environment for development and technical implementation of data collection as well as the evaluation of closed-loop solutions. The studies are presented within the clinical context, covering the necessary information regarding their approvals, conduction as well as the performed surgical and medical procedures. Additionally, a brief outlook to the relevant technical aspects regarding data collection and automation in relation to the individual studies is given.

4.1 Applications and Approvals

Both studies were approved by the local animal experimental committee in accordance to the National Guidelines for Animal Care and Handling and reported to the animal protection commission at Regierungspräsidium Tübingen (Germany) by the animal protection commissioner of Tübingen University according to applicable guidelines and regulations (German animal protection law (TierSchG) §7a). Care and treatment were performed under guidance of a veterinarian in accordance to the national and European animal protection laws to prevent animal harm. Study animals were kept in standardized conditions with fasting at the operation day, water was available ad libitum. All animals were sacrificed in deep general anesthesia after the permitted duration in terminally trials classified as non-recovery according to the European guidelines (2010/63/EU [44]).

4.1.1 Study Setting

An overview of the final study setting is provided in Figure 4.1. General monitoring was performed with a Philips MP50 monitor, Osypka ICON and Pulsion PulsioFlex. For ventilation a Dräger Evita XL was used. Infusions were administered with infusion pumps of the B.Braun Space series. Body and urine weight were measured with in-house developed scales. Temperature management was performed with a heat mat and post-operative with a 3M BairHugger forced-air warming device. Laboratory analysis was performed with a Radiometer APL800 Flex BGA analyzer and ACT was measured with a Medtronic ACT Plus device.

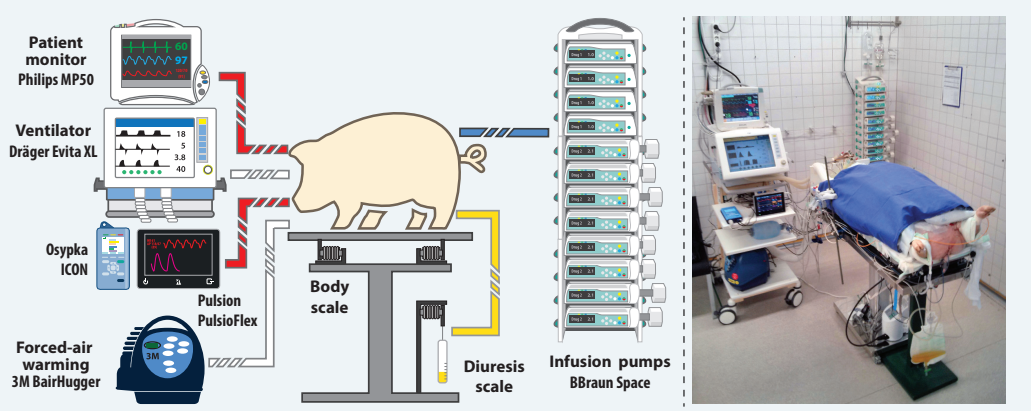


Figure 4.1: Schematic illustration of the fully equipped study setting (left) and photo (right). A central control interface and laboratory equipment are not included in this figure.

4.2 Common Protocols & Procedures

4.2.1 Pre-operative protocol

Sedation of the pigs was performed by intramuscular injection of atropine 0.05 mg/kg BW and azaperone 4.0 mg/kg BW. Narcosis was introduced about 20 minutes later by intramuscular injection of midazolam 0.2 mg/kg BW and ketamine 14 mg/kg BW. After another 10 minutes, tracheal intubation and an injection of up to 100 mg propofol was performed by a veterinarian. Ventilation was performed in a volume-controlled mode with an enhanced O_2 level of 40 percent to maintain sufficient oxygen saturation. General anesthesia was then maintained by TIVA of ketamine, midazolam and fentanyl through an IV access at the ear. Patient monitoring for ECG, oxygen saturation, and temperature was established.

4.2.2 Surgical Procedures

For venous access, a 5-lumen catheter was inserted in the *vena jugularis* by ultrasound guidance (Seldinger technique) or if necessary in open surgery by a physician. Thereafter, an arterial catheter with PiCCO temperature measurement was inserted into the *aorta* through punctuation of the *arteria sapherena* by the same means. This provided arterial and venous access for infusions and collection of blood samples and established blood pressure and hemodynamic monitoring. Finally, a suprapubic urinary catheter was placed by minimal laparotomy.

4.2.3 Post-operative & observation protocol

In the post-operative phase, the animal was given time for physiological stabilization and used for connection and calibration of all required medical devices and measurement setups. Therapeutic anticoagulation was performed as a thrombosis prevention with unfractionated heparin. Prophylactic antibiotics (ceftriaxone 2 g/d) were administered daily. Observation and research were performed as described in the protocols of the individual studies. Supervision by physicians and animal study investigators for observation of vital functions and anesthesia was present at all times.

4.2.4 Post-observation protocol

After the permitted study and observation times, the animals were sacrificed with an intravenous injection of embutramide (T61) and general pathological findings were noted. An autopsy of selected tissue samples was performed postmortem if any indications or complications occurred during the trial.

5 Study details

5.1 Autopilot Study (AP)

This first study, Autopilot (AP), consisting of research on ten porcine animal subjects (ap001–ap010), was conducted in the years 2013 to 2015 and approved with application C2/13 with the title *"Entwicklungsbegleitende Evaluation von Medizinprodukten in Terminalversuchen"* (development-related evaluation of medical devices in terminal animal trials). For each animal subject an observation period of 72 hours was admitted. The scope of this study included the medical evaluation of various devices and the integration into a unified framework was established alongside those tests. Beside this evaluation of commercially available devices, measurement setups, e.g., for diuresis were custom build and tested. Fundamental development and testing of the software framework, described in the next part of this thesis, was performed and used to collect clinical data from an increasing number of sensors and sources. This allowed for and guided the integration of more and more automated procedures within the experimental clinical setting. Research for the first animal subjects (ap001-ap004) was performed to establish data collection of individual devices and development of device interfaces to guide the overall framework development. From animal subject ap005 onward, the framework was fully functional in a first version and all data could be collected within a unified database.

5.1.1 Clinical Questions

During this study several clinical questions were researched. This included, the evaluation and comparison of different state-of-the-art medical devices and finding practical and clinically relevant aspects subject to automation. Vital data for changing conditions was obtained as the pigs were bedded in different positions (left, right, back) every few hours.

5.1.2 Technical aspects

From a technical viewpoint, this study was used for the development of the medical framework and integration of individual devices. Therefore, the communication protocols for ventilator (Dräger Evita XL), patient monitor (Philips MP 50), and infusion pumps (B.Braun Space) were analyzed and the devices were integrated as individual components of the framework. Ventilation device and infusion pumps were configured with remote control capabilities and could be used for automation of different aspects like CO_2 management by adaption of RR and management of various infusion rates. From animal subject ap002 onward, glucose and calcium control was automated using a developed fuzzy logic controller. A diuresis scale for measuring of accumulated fluid within a urine bag was developed and integrated.

5.2 Volume-Need-Analysis Study (VNA)

The second study, Volume Need Analysis (VNA), was dedicated to the clinical research question of assessing the volume state of patients and conducted on ten individual animal subjects (vna001-vna010) in the years 2016 and 2017 under application C1/16 *"Evaluation eines automatisierbaren Indikators für den Volumenstatus"* (evaluation of a volume state indicator that can be automated) with an admitted duration of 96 hours for each individual animal subject. Guided by the observations of changes in the volume state of the individual pigs regarding their bedding position in the data collected by the developed software framework in the AP study, the question of assessing a reliable volume state of the animals became of clinical interest as preliminary research showed insufficient knowledge regarding assessment of the volume state in clinical practice for hospitalized patients. As development of the framework was finished during the AP study, data collection for analysis could be performed for all animal subjects of this study. Thus, given a comprehensive set of reliable collected data, yet unknown relations between the volume state and other physiological parameters might be detected and guide further clinical research regarding this important topic. To obtain almost complete knowledge about the volume state of the studies animals, information about all infusion rates and diuresis as well as the body weight of the pigs was continuously monitored. To establish reproducible conditions, the developed automation using fuzzy controls was utilized for management of homeostasis. During the study and this management, the topic of better temperature control became of interest and usage of forced-air warming blankets was implemented and suspect to research regarding collection of data, control and closed-loop management from animal subject vna005 onward.

5.2.1 Clinical Questions

The clinical questions regarding this study were the collection of research data aiding the assessment of patient's volume states according to the S3 guidelines with methods and protocols proposed and recommended by the Arbeitsgemeinschaft der Wissenschaftlichen Medizinischen Fachgemeinschaften (AWMF). Therefore, all vital signs and data regarding the volume state were collected with the developed software framework and are subject to further analysis, exceeding the scope of this thesis. For providing a broad variety of research data, the animals were given defined amounts of fluid by IV infusions to reach an increased fluid state. Thereafter, fluid supplementation was stopped and the body fluid was reduced by usage of diuretics. Assessment of the fluid state was performed at defined volume states by performing a variety of tests, including a Trendelenburg maneuver, a respiratory hold test, laboratory analysis, and ultrasound measurements of the diameter of the *vena cava*. Due to the mass distribution in pigs, instead of raising legs during the Trendelenburg maneuver, the entire pig was tilted 20-25 degrees head-down. The respiratory hold was performed after the inspiration cycle, allowing for the pressure and volume of the lung to displace liquid in the thoracic cavity, presumably having a similar effect on the volume state as the leg raise test.

5.2.2 Technical aspects

With the framework fully functional at the beginning of this study, the integration of additional devices and the development of further closed-loop systems was the main research focus. A measuring setup using an array of load cells for continuous body weight monitoring was developed and integrated into the framework. Automated homeostasis management using the developed fuzzy logic controllers was utilized for all animal subjects. Hemostasis management was automated by integration of an ACT measurement device and implementation of closed-loop for a heparin infusion pump. Additionally, besides numerical data collected every second, the interfaces to patient monitor and ventilation device were expanded for collecting high-frequency waveforms of ECG and respiratory parameters. Using temperature management from animal subject vna005 onward, the forced-air warming device was extended with remote control capabilities and automated closed-loop control was established for animal subject vna007 to vna010.

5.3 Study overview

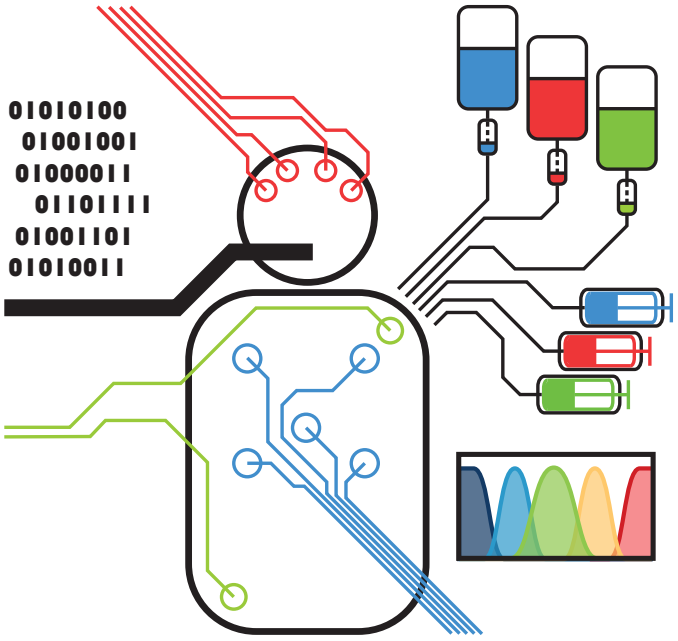
A general overview of the porcine subjects of the two conducted studies with details regarding gender, weight, study ID, and study application are given in Table 5.1 as a reference for subsequent parts of this thesis. In total 20 DL pigs were used for the studies. All animal subjects were female with an average body weight of 44.1 ± 1.5 SEM. The total observation time of 1373 hours is composed of 500 hours for the AP study and 873 hours for the VNA study with mean observation times of 50 and 87.3 hours, respectively.

Table 5.1: Overview of the individual porcine subjects with study application.

animal subject	duration [h]	gender	weight [kg]	application
ap001	24	♀	43.5	C2/13
ap002	24	♀	44.8	C2/13
ap003	74	♀	30.5	C2/13
ap004	65	♀	37.8	C2/13
ap005	72	♀	44.8	C2/13
ap006	36	♀	30.6	C2/13
ap007	72	♀	55.4	C2/13
ap008	52	♀	51.5	C2/13
ap009	68	♀	49.4	C2/13
ap010	13	♀	47.6	C2/13
vna001	96	♀	40.8	C1/16
vna002	57	♀	46.5	C1/16
vna003	96	♀	42.0	C1/16
vna004	96	♀	48.5	C1/16
vna005	96	♀	38.6	C1/16
vna006	96	♀	51.1	C1/16
vna007	96	♀	43.6	C1/16
vna008	96	♀	55.0	C1/16
vna009	96	♀	41.3	C1/16
vna010	48	♀	39.0	C1/16
	Σ 1373		∅ 44.1	

Part III

Experimental Medical Monitoring and Control Framework



6 Motivation

To perform data analysis and machine learning, the essential first step is acquiring a sufficient amount of data for the analysis. Unfortunately, in the field of healthcare and medical informatics data is hardly available. Public databases like MIMIC [106–108] only provide subsets of vital data and are not sufficient for a full-scale analysis and development of closed-loop systems. Therefore, gathering of the needed data was the uttermost important task to be performed prior to any subsequent analysis. Currently, most medical devices provide only manufacturer-specific protocols or are designed for billing and quality assurance purposes only, while lacking the required data resolution for exporting research data. This provides difficulties for matching of individual measurements from different devices within the time domain.
The collection of data to aid better understanding of interactions and the development of better models is also an important step for further research in animal studies. It has significant potential for reducing required animal studies by simulation with sufficient models. Furthermore, given a working system, the concept for clinical applications is not limited to theory anymore, but can be shown in a practical setting used and evaluated by physicians. This is a very relevant aspect as such systems will be used by physicians and nurses in the hospital and thus, an early feedback from those users may provide important information and guide further development.
Unfortunately, no medical framework for the task of data collection and testing various soft- and hardware concepts regarding automation and closed-loop controls in an experimental clinical environment exists. Only independent solutions for individual aspects are available for some specific tasks and devices. Using an existing framework from other research domains like ROS may be possible. However, due to a mixed Linux and Microsoft Windows environment with dependencies on manufacturer-specific drivers and libraries, usage and adaption of a mainly Linux-based solution could lead to high amount of work for integrating those required devices if possible at all. Therefore, trying to adapt such an existing framework to the needs of the medical environment was ruled out due the uncertainty of compatibility and the goal of avoiding limitation to a single operating system or platform in the heterogeneous landscape of medical devices and systems.
Still left with the fundamental problem of connecting medical devices in a scalable, interchangeable, and not manufacturer-specific way, the presented framework was developed as an essential part for automation and data collection covered in subsequent parts of thesis and partly published with its fundamental concept [191]. The following part with its chapters presents the developed concepts in depth, including the physical implementation at the Institute of Experimental Surgery at the UKT. Finally, the evaluation during the conducted porcine studies outlined in the previous part is presented from a technical viewpoint, including the amount of collected data and the achieved data throughput.

6.1 Framework Requirements

Given the freedom of developing a novel framework for interconnection of medical devices within an experimental medical setting, requirements and needs for such a system had to be chosen in the best possible way to facilitate the desired goals for data collection and automation. Those fundamental requirements are:

First, the framework must be capable of processing and logging arbitrary information from all devices connected to it and transmit the information between any two devices. As automation in the goal, providing only a data collection framework is not sufficient but information must be distributed between the devices without limiting the available communication paths.

Second, the framework and its applications must be able to handle arbitrary information provided by various interfaces and information formats of the available medical devices, still lacking standardization. Thus, interfaces to each device must be developed individually and integrated into the common framework.

Third, all information, once gathered from the individual devices using the respective and specific interfaces, should be processed and transmitted in a standardized format within the framework to allow easy expansion, interaction, and storage.

Fourth, information from all devices should be available for all connected devices within the framework on time, therefore allowing algorithms and other programs to process the data and perform analysis as well as adaptions of the medical devices to truly achieve an online closed-loop system.

Fifth, no distributed solution with many devices, individual logging, or transmitting all information independently should be used as this would require retrospective collection and matching of the individual data sets and maintain data dispersion. Thus, all data should directly be stored in a single central database with a common format for further processing.

Sixth, all messages and devices should maintain a homogenous time base to allow a matching of the obtained results, allowing for the analysis of causalities and closed-loop controls. Thus, a loosely connected system involving various individual laptops would be prone to errors due to differences in the systems' clocks. A retrospective collection and matching of the obtained data may therefore hardly possible. To solve this problem, a common framework with time synchronization is therefore required.

Seventh, the system should be modular and scalable for multiple patients and beds, while sharing centralized systems like the one used for BGA. This especially provides difficulties in the hardware and network domain, where separation of the individual patient's systems to avoid interferences and crosstalk between numerous medical devices may become relevant.

7 Design and Concept

The developed software framework for collection of medical data and control of medical devices in an intensive care setting is designed for bidirectional transmission of any type of data including measurement results and control commands. Using a client-server concept, a controlled interaction between the individual devices can be established while maintaining an overview of the current setting. Using a server as a message relay, all transmitted information can easily be logged and collected for retrospective analysis. The presented framework is designed as a modular research system where the ability for expansion and improvements was the uttermost important aspect. Other aspects like security, redundancy, or error handling during the transmission process were only subordinate goals. However, those goals were still considered during development of the framework, yet functional implementation would require additional expansions and further testing. To achieve this modularity and expandability, a plugin-based concept, allowing for extension or replacement of single components and providing re-usability of already implemented features, was chosen. This re-usability is an important aspect as it allows for faster development. Furthermore, it allows for better testing of the individual reusable components due to a larger number of test cases for the detection of flaws and bugs. Once such problems are detected, they can easily be fixed for all usages within the framework at once. This is especially useful for critical applications like network communication or encoding and decoding of messages, utilizing a single implemented and reused plugin.
In this chapter, an overview of the plugin architecture and its features is given by introducing the individual components. Then, interactions between those plugins and interaction between different applications within the framework are presented, including the processing of messages using filter rules. Furthermore, aspects like the used database structure and configuration of the individual applications are briefly highlighted.

7.1 Plug-in Architecture

The framework is developed using a plugin architecture for rapid development and integration of additional features, modularity, and reduction of redundancy. Therefore, a core application with a variety of different plugin types was developed. An overview of the entire architecture with the different types of plugins and general interactions is shown in Figure 7.1. The details of each plugin and the application core are described in the following sections.

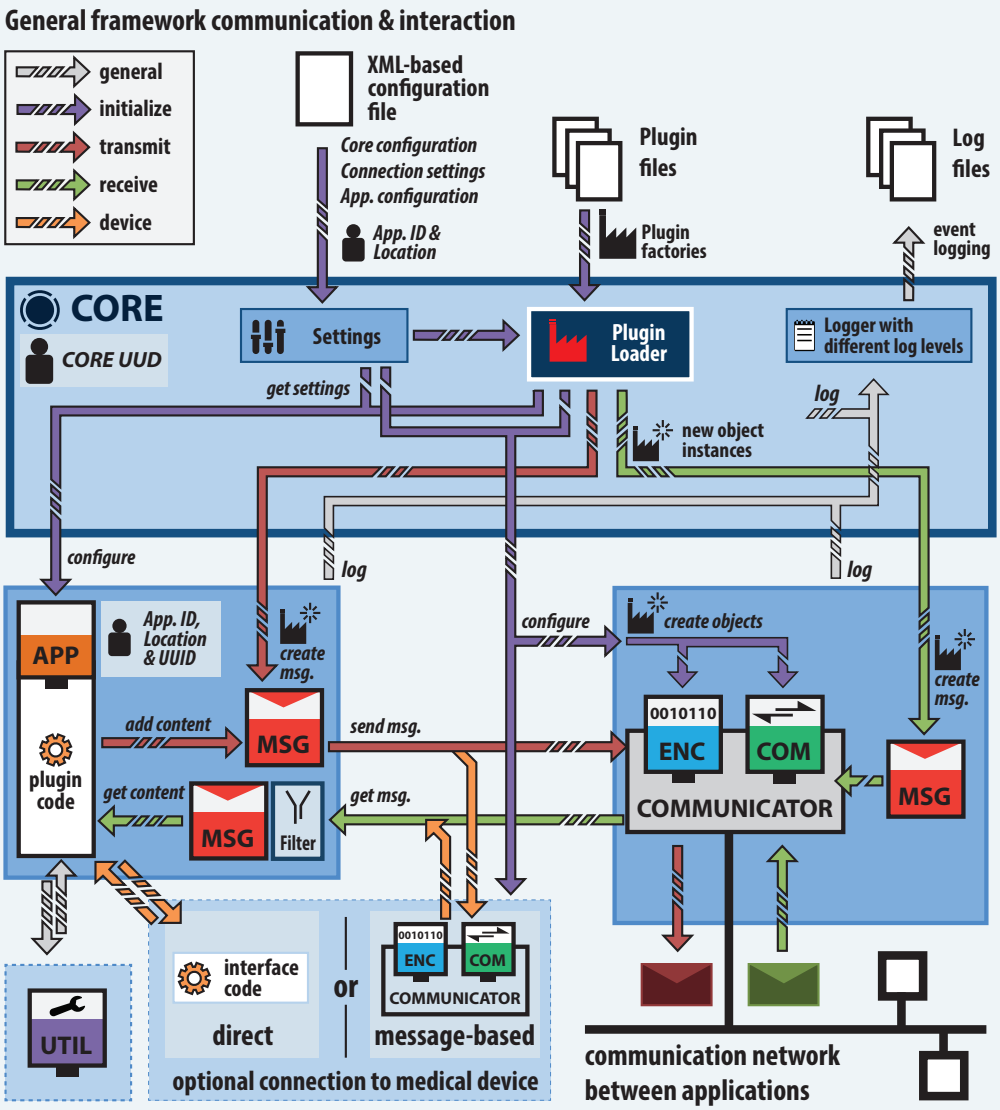


Figure 7.1: Overview of the plugin architecture. For initialization, the settings are read from an XML-based file and used for configuration of the application core with a plugin loader and logger. The class objects for application and communication plugins are then crated from factory objects loaded from individual dynamic libraries. Message-based communication is performed using objects loaded from message plugins. For transmission, these plugins are set-up by the application plugin and processed by a Communicator object with an encoding (ENC) and communication (COM) plugin. To receive messages, this process is reversed. Communication with medical devices can be performed using this message-based system of through direct communication implemented within the individual plugin. Additionally, utility (UTIL) plugins can be loaded and used for encapsulation of specific, reusable features.

7.1.1 Core application

The central building block of the software framework and plugin architecture is the core application. This executable contains the base functionality and performs a variety of general tasks. First, it is used to load a XML-based configuration file that contains further information about the plugins and settings to be utilized. Second, it provides a central logging of status and debug messages for all plugins within the framework. Logged data is reported in a console window and written in a log-file based on an error reporting level that can be set at the start of the core application with a run-time parameter. There are several levels reaching from trace, e.g., containing dumps for array elements, over debug information and program states to warnings and errors. The core application scans the available plugins and provides them for usage by loading the dynamic libraries and interacting with the plugin factories to create plugin instances when requested. Additionally, all physical network or bus connections to be used by the plugins, e.g., via Ethernet or serial port, are initialized by the core application. For identification of the running core application, the program provides a unique identifier (UUID) of the running application, which among other applications is used for tracking the client at a reconnect to the server.

7.1.2 Plugins

Plugins are used to provide additional and specific functionality for performing various defined tasks within the framework. Plugins are loaded and maintained by the core application and defined in the XML-based configuration file. Each plugin is stored as a platform-specific library developed in C++, utilizing the plugin interfaces provided by the Qt framework. For easy development and integration, each plugin consists of a fixed base structure and well-defined interfaces to the application. To allow attachment to the core application, each plugin implements a factory pattern for initializing individual instances of the plugin. Each plugin has a unique plugin name that is compiled into the application and used in all locations where the plugin needs to be referenced by its name. Depending on the function, which can be inherited from a few different basic types (application, communication, message, utility, encoding/decoding), specific templates and abstract classes with interfaces are implemented to allow seamless integration within the framework. Each plugin provides its plugin name and plugin type and has common methods for initializing, starting, and stopping that can be called by the core application. Interaction with the core and other plugins is implemented using Qt's signal and slot concept. For communication within the framework, information is passed through those signal- and slot interfaces using messages, or more specific, message objects that are loaded from specific message plugins. This provides an abstraction layer between the information to be transported and the message handling process itself. As used messages objects are stored as smart pointers, this allows automated destruction if the reference count reaches zero, indicating the message object is not needed anymore. In what follows, a short overview of the different plugin types and their usage will be provided.

7.1.2.1 Application plugins (APP)

Application plugins of the developed framework integrate individual program functionality and rely on obtained and sent messages for communication and interaction. Each application plugin can provide a graphical user interface (GUI) if needed. If the plugin is integrated into a graphical environment, the core application can request and show the GUI in an appropriate manner. This is especially useful, if multiple plugins are loaded (e.g., in a multi-plugin application as described in section 7.5), where many plugins are integrated into a common graphical environment. The individual parameters and settings required by the plugin are read from the XML-based configuration file and used for the configuration of the plugin, e.g., by defining constants, parameters, IP addresses, scaling factors, or other required settings. Most APP-plugin are inherited from an abstract base class, providing basic functions like handling of configuration parameters and message processing. In each APP-plugin, filter rules for the messages to be received and processed by the plugin are provided. If some functions are required by different applications, they can be outsourced to utility plugins and accessed by the application plugin to limit code duplication.

Abstract application client The abstract application client is a fundamental base for all client plugins and used for methods and functions that are required in all client application plugins to reduce functional redundancy and code duplications. It implements many general and fundamental methods for communication with the core and interactions within the framework. Those functions include management of the connection to a central server within the framework, sending alive beats, and pre-processing messages with filter rules. On application start-up, the required parameters within a XML-based configuration file are parsed by this base class and accessible by the APP-plugin. Additionally, required parameters can be defined and are ensured to be present in the configuration file on program start.

7.1.2.2 Communication plugins (COM)

Communication plugins are used to provide an abstraction layer between the message-based objects within the framework and the physical communication over the chosen transport medium. They provide basic functionality for establishing communication channels, e.g., by opening serial ports or TCP connections, maintaining the connection, and handling errors. Data processing is performed on raw strings or byte arrays without further knowledge of the message context. Messages are padded with delimiters and sent over the chosen physical interface. For example, by using a COM-plugin implementing the Ethernet protocol, messages are sent as TCP packages using defined start and stop characters. Alternatively, by using another plugin, message objects or direct byte commands can be encapsulated and transmitted by serial communication. If a COM-plugin receives data from the physical data stream, the obtained information is buffered and automatically parsed. If the defined delimiters are detected, individual messages are extracted and a signal indicating the availability of a new message is emitted for further processing by APP-plugins. For sending messages, APP-plugins emit message objects which are processed, serialized, and then transmitted by the COM-plugin. Those

intermediate steps during transmission and reception of messages are performed by other plugins and modules like the encoding/decoding plugin, described next.

7.1.2.3 Encoding/Decoding plugins (ENC)

Encoding/Decoding plugins are used for structuring the information stored in message objects any may provide secure encryption for data transport. An example for such a usage would be manufacturer specific commands to be exchanged between different devices of this very manufacturer, without providing information to the framework, central database, or other devices. The information would be processed and distributed encrypted only and if connected to a central database stored encrypted as well. For debugging purposes or other analytical tasks, the stored and encrypted information could be decrypted by the manufacturer at any time. For communication, message objects need to be processed and embedded into a well-defined serial data structure. Between different framework applications, communication is performed over Ethernet using an ENC-Plugin with XML-encoding to store and transmit the required information within a defined XML-schema. Other encoding standards required for communication with specific device, like HL7, are provided by additional ENC-plugins.

For sending a message, the message object is serialized by the specific methods provided in the appropriate message plugin and embedded in the general structure generated by the ENC-plugin. This general structure includes header information, containing details of the sender and designated receivers, a unique message identifier, and the message type.

Received messages are processed by parsing the raw data to obtain this universal header information. As the message type is included, the receiver can determine the required message plugin. Further processing of the message content is then performed by this plugin, resulting in the desired message object that can exchanged within the application. If the message type is unknown to the receiver, the message is automatically discarded.

Besides the message-specific processing, commands for interaction between clients and server, like ACK messages or settings, are directly implemented in the ENC-plugin.

7.1.2.4 Message plugins (MSG)

Message plugins are the most essential plugin type and used for information exchange within the framework by message objects. They can be used to define arbitrary message objects and must provide methods for serializing the stored information to be encoded and transported over the framework. For interaction with other parts of the framework, especially the APP-plugins, they have appropriate setter and getter methods for handling the stored information. As specific serialization and deserialization is implemented within the plugin, only the sender and receiver of a message need to have access to the appropriate MSG-plugin for handling the message. Other devices, may just process the header information and discard the message if no appropriate plugin is available. This allows for easy expansion of the framework and limits the overhead of plugins at each device. As already implied in the description of the plugin architecture and the other plugin types, message information is first obtained by the COM-plugin, processed by the ENC-plugin, and finally decoded and converted to a message object by the MSG-plugin. Using Qt's signal-slot concept, the message object is then provided by emitting

an event to the receiver's APP-plugin. The APP-plugin can fetch this message object for further processing. The individual instances of the message objects are referenced as smart pointers within the framework and automatically deleted if the message information is no longer needed. Whereas for receiving a message, the ENC-plugin obtains a message object instance from the factory of the MSG-plugin, a new object instance is obtained by the APP-plugin for sending. Information then is added by accessing the specific setter methods of the MSG-object. Once all information is included, the message object is pushed into the send pipeline where ENC- and COM-plugins perform further processing as already described.

Message abstraction layer In general, all information can be stored as a key-value pair. As all messages need to be serialized for transmission and de-serialized at the receiver, such a general approach allows for an easy and homogenous processing, especially when using a XML-based structure. However, by using such a general strategy, only generic getters and setters can be provided. With such a high level of abstraction, the only guidance for usage of messages would be a sufficient documentation of the message type and the valid key-value pairs. Error handling for each get and set operation with possible additional or missing keys would be required. Therefore, a better approach is to maintain name-specific getters and setters for the different types of message plugins as described earlier to avoid such possible mix-ups and provide separation of message types. Thus, the described high abstraction level with a key-value structure, denoted as the general message data format (MDF), is only used as a foundation for the individual message plugins and inherited by them as the underlying message structure. The individual message plugins provide a defined interface with named getters and setters for the individual key-value pairs, therefore allowing the developer to easily gain self-explanatory knowledge about the information that can be obtained from or stored within a message. This promotes easy validation and reduces coding errors. Hence, providing a tradeoff between generalization with reduced overhead and required work on one hand and maintaining self-explanatory code on the other. Additionally, embedding of the key-value functionality in specific getters and setters of the individual plugins allows the encapsulation of the specific methods for storing, retrieving, and validation of the specific key-value pairs within the plugin itself, providing further reduction of dependencies and better encapsulation of the code.

7.1.2.5 Utility Plug-ins (UTIL)

Utility plugins are an additional and optional type of plugins for general purpose encapsulation of code to reduce redundancy. Whereas other plugins have defined interfaces and are mandatory for framework application and communication, UTIL-plugins and can be used in a more general way.

7.2 Identification of messages and applications

As already outlined within the description of the plugins, each core, application, and message provides unique identifiers for processing and filtering of information within the framework. This information is transmitted among each message and used for routing by the server. Additionally, this information can be stored with the message content in a database, allowing for complete traceability of each interaction. As identifiers are essential for message filtering, described in subsequent sections, a brief overview of the different identifiers is now provided:

Application identifier

Identification of each running application within the framework is performed by an application identifier for the core application and the running plugin. Each one is unique and can be used to identify specific program instances or linkage of individual applications at run-time. Additionally, this unique identification is used for handling client connections to the server, including the detection of newly connected or reconnected applications, and for message processing.

Patient identifier

A patient (or animal subject) identifier is used for identification of the current patient. Each device that can be associated with a single patient is assigned their unique patient ID. The ID is used for filtering patient-specific messages within applications. For the database, where the information of multiple patients is stored, it provides the necessary information to assign the received data to the correct patient-specific database table.

Location identifier

Besides identifiers for applications and patients, additional identification can be performed by the location of the used devices. It is useful for devices or applications in a fixed setting like the bedside ventilator, patient monitor, scales, and infusion pumps. As such devices often remain stationary and linked to each other in hospital settings, a new patient ID can easily be set or updated for a group of devices. Given a common location, this can even be performed if different patient IDs were previously set, otherwise requiring more complex updates to reach each device within the group by means of other identifiers.

Device identifier

Another identifier is provided by setting a unique name for an application or device. Besides the name of the running plugin and the automatically generated application ID, this name can be set within the XML-based configuration file. This allows unambiguously device identification independently from assigned patient, location, or running program instance. It is especially useful if devices should always be kept paired, only listening to each other's messages, independently of similar messages exchanged between other devices.

Of course, all those presented identifiers may be used in combination, e.g., as a safety feature to assure that a specific patient is assigned to the correct bed. This is especially important for automated controls following a therapeutic scheme, where mix-ups could

have fatal consequences. An illustration of the combinations and interactions with individual identifiers is shown in Figure 7.2.

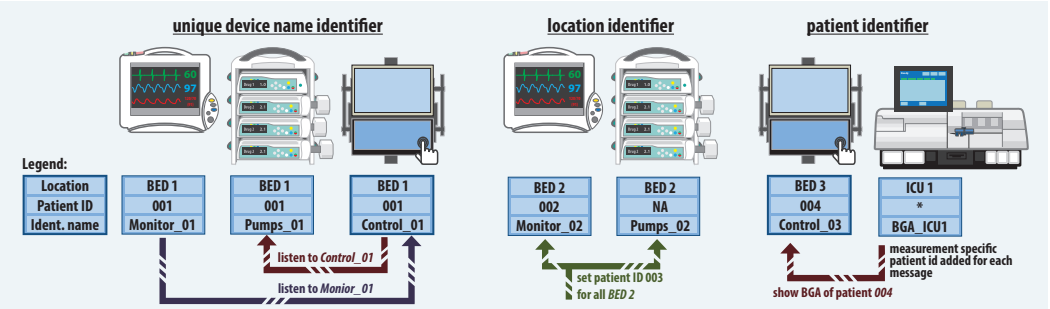


Figure 7.2: Illustration of the usage for different identifiers. Control loops can be set to interact with specific other devices independent of the assigned patient (left). Location identifiers can be used for grouping devices and applying commands, like setting a new patient ID for a specific bed (middle). Patient identifiers can be used to filter messages associated to a specific patient (right). As some devices are used for multiple patients, the individual patient ID is added on a message-by-message basis.

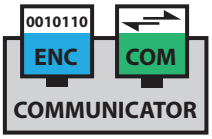
7.3 Message-based communication

Communication within the framework is based on transmission of individual messages between the individual applications. The individual components and plugins required for this communication were already described and a brief overview of the implemented message-based communication was given. To provide deeper insight, the detailed communication flow and message processing with filtering will be described in the following sections. First, an additional helper structure and abstraction layer for managing the interaction between the described plugins and the core needs to be introduced. Then, the message flow with filtering rules is described.

7.3.1 Connections

For processing of messages to be sent or received, the interaction between a pair of COM- and ENC-plugins is essential within the developed framework. As the combination of these two plugins is always required, a further abstraction for this relationship is used. A Connection describes a specific communication channel for sending and receiving the information by defining the names of the used ENC- and COM-plugin and, depending on the connection type, the specific settings like IP addresses and ports. Each Connection is defined within the application's XML-based configuration file and has a unique name for referring to this channel during runtime. At the start of the application, all defined connections are loaded by the core. Based on those definitions, the communication channels are established by creating a Communicator object within the application core for each defined Connection.

7.3.2 Communicators



Communicator objects are the implementation of this additional abstraction layer for message processing within the framework. They are the actual interfaces for communication through a specific channel like Ethernet using XML-encoded messages. Each Communicator is created by the application core based on its Connection settings, defined within the XML-based configuration file. As the Connections and Communicators have a unique name, applications can easily select the desired communication channel at run-time for each message to be transmitted. Each Communicator provides an interface for sending messages and is connected to the APP-plugins via Qt's signal-slot system. Thus, automatic notification and processing of received messages by the application can be performed. Additionally, the Communicator provides interfaces and signals for the detection of new clients or disconnects. This is especially useful for the server application but also used by application plugins to stop creating new messages if the connection to the server is closed. Transmission is only continued once the connection is re-established.
Furthermore, the Communicator provides a message buffer for automatic resending and handles the acknowledgment process for successful message transmission. Once a message is sent, the sender stores the message and the message UUID until verification is received. Verification is performed by the receiving Communicator by returning an ACK message with the UUID of the message. Upon reception, the sender discards the corresponding message backup. If no ACK is received within a defined time frame, the Communicator tries to resend the message for a defined number of times before finally discarding it and reporting an error. A detailed illustration of the message flow within the Communicator, utilizing the ENC- and COM-plugins is shown in Figure 7.3.

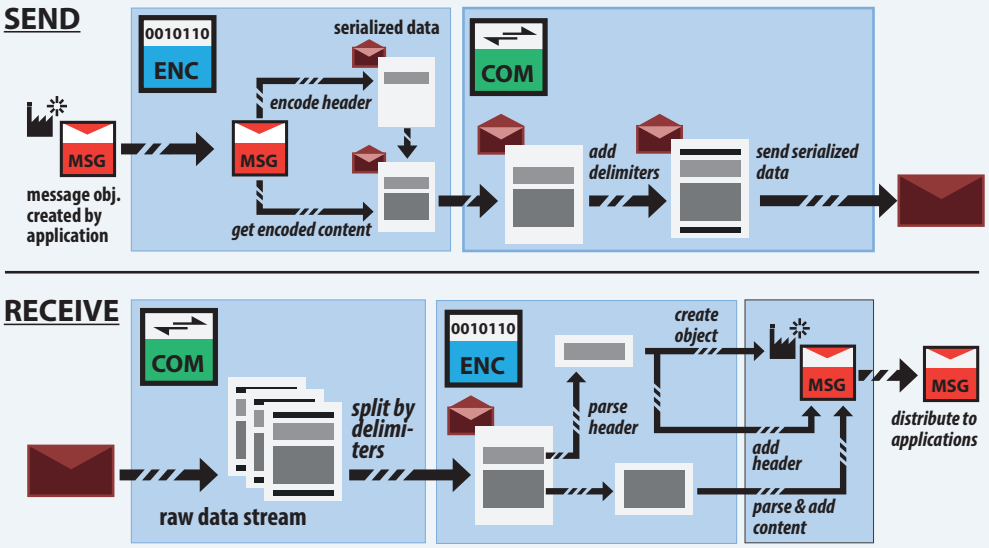


Figure 7.3: Illustration of the message flow of message objects through a Communicator utilizing ENC- and COM- plugins for sending and receiving a message.

7.3.3 Message-flow and filtering

As the entire communication within the software framework is based on massage objects, messages need to be identified and filtered for distribution and processing. This filtering may be performed by broad variety of different criteria, for example, regarding messages belonging to a specific study or a specific message type. To allow for such a filtering, each message header contains information regarding to the message type and an optional subtype. The message type is defined by the message plugin, providing the appropriate methods for handling the message and the contained information. The additional subtype can be defined if different data should be distinguished but inherits the same data structure and information. An example for such a scenario is the measurement of blood gases. The BGA results for arterial and venous blood samples are processed and distributed as the same message type but with different subtypes. Because these two results have very different characteristics, e.g., in the oxygenation levels, they must not be confused in further processing. Therefore, filtering based on the subtype is an important feature. To allow general filtering based on the different characteristics and identifiers for messages and applications, the various filters need to be defined and processed. Those filter concepts and the processing will be described in the following.

7.3.4 Filters

Filters are used for the management of information flow within the framework. Filter rules can be created for different categories like patient ID, message types, subtypes, UUIDs, and identifiers. For transmission of information to specific application instance only, device identifiers with a unique device name may be used. Specific applications can be selected by application identifiers and location filters, e.g., to only receiving messages from a specific defined location like a hospital bed. Beside the combination of different filters for the various identifiers, each filter rule can be inverted. This allows the exclusion of one or more conditions and the definition of universal filters. To describe the filter concept in more detail, a formal description should be given:
Each filter $F = r \in R$ is a set of filter rules, where R is the space of all possible filter rules. A filter rule $r = (A, inv, t) \in R$ is a tuple, where $A = \{S_1, \ldots S_n\}, n = |A|$ is a set of strings $S = (c_1, \ldots, c_n), n = |S|$, where c is a character. $inv \in \{0,1\}$ is a flag indicating if the set is an inclusive or exclusive description and $t \in \{type, subtype, UUID, deviceID, appID, location\}$ is the filter type.
Using the combination of multiple filters with freely definable rules, all possible and specific combinations for message filters can be defined.

7.3.4.1 Filter comparison

Within each application plugin, an individual set of filters can be defined. It is used for comparison with the characteristics of each received message to determine if further processing is needed or if the message can be ignored. As each message provides the identifiers of the source application and the details of the message in the universal header, comparison to the filter rules can be performed for any message object processed by the Communicator.
The filtering step is performed by comparison of the message characteristics to the filter

rules in a top-down fashion. If a first filter match occurs, the message is accepted and will be further processed by the application. Only if the message has passed through all filter comparisons without a match, the message is discarded. For comparison to a filter, the header information of a message is stored within a filter structure and compared.
A filter rule, e.g., a message characteristic, $r \in R$ is matched to another filter rule $s \in R$, if their types match $r_t = s_t$ and depending on the inversion rule s_{inv} of the filter, a match $m \in 0,1$ is an indicator function with

$$1_m(r,s) = \begin{cases} \exists\ r_i \in r_A, s_j \in s_A : r_i \leftrightarrow s_j & \text{for } s_{inv} = 0 \\ \nexists\ r_i \in r_A, s_j \in s_A : r_i \leftrightarrow s_j & \text{for } s_{inv} = 1 \end{cases}$$

The matches are calculated for each identifier of the filter rule and once a match is found, the message is accepted by the client. Otherwise, the comparison is continued with the other filter rules and additional filters.

7.4 Client-Server structure & interaction

The client server structure in the developed message framework provides the foundation for controlled message transport. Whereas devices could interact directly with each other in small networks, e.g., by defining one application as a server and the other one as a client, in a setting with multiple devices and many interactions or even multiple patients processed in parallel, the large number of transmitted messages would flood the network and cause unnecessary load and delays for handling and filtering required by each device. Thus, it is reasonable and beneficial for messages to be transmitted only to devices for which the message is relevant. This may be performed by a peer-to-peer network with device identification. Alternatively, as mentioned in the introduction regarding the scientific purpose and the evaluation of such a system, it is very important and useful to have an overview of all interactions and messages that are transmitted. Therefore, a central server with logging capabilities for all information in a central database is an important feature. This client-server concept is implemented within the described software framework with the client as well as the server being APP-plugins. The only difference, aside from APP-plugins itself, is the configuration of the COM-plugin: Configured for running in server mode and client mode, respectively. An illustration of the typical interaction between a client and the server is shown in Figure 7.4 and briefly described in the following.

7.4.1 Server

The server is an APP-plugin that provides the necessary structures for handling multiple clients. Additionally, it provides status messages for the state of the system, e.g., which clients are connected. The server application manages client connections and stores the login information of the clients. This information includes the client's filter settings to be used. As the server is aware of those filter rules, each message is processed at the server and distributed only to clients, where running APP-plugins are interested in the message content. Those filter rules can be updated by the client at any time by sending a filter update

message. This, for example, will often be performed if the server sends a study update request to change the patient ID of the current application. If the client is configured to filter messages according to the patient ID, new filter rules are required and must be announced to the server.

7.4.2 Client

Clients are all applications connected to the server via a Communicator using a COM-plugin configured in client mode. Clients connect to the server at program start and exchange their information, e.g., the identifiers, XML-configuration, and the filter rules with the server. The server is then able to identify the clients to forward those messages the client is interested in. To verify the running condition of each client and the network state, every few seconds the communicators of each client send special alive messages as their functional vital sign to the server.

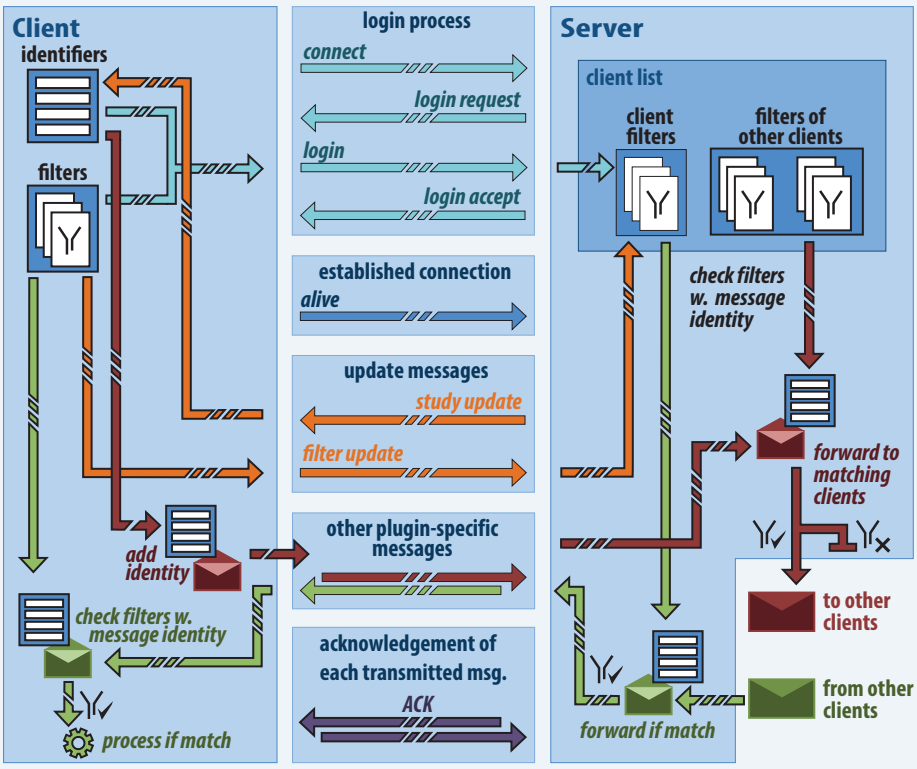


Figure 7.4: Client-server interaction with the different message types and the performed filtering processes.

7.5 Multi-plugin applications

For using a single computer with a multitude of different apps, the software could be started several times. However, starting multiple Cores and Communicators, connecting to the same server will cause an unnecessary overhead. Dealing with a network connection over Ethernet, multiple communication channels will not cause a problem, as the connections can be established to the server over different sockets and ports. However, using other transport media like bus systems, e.g., CAN, a client computer can only establish a single connection to the server. Otherwise, multiple parallel connection cables would have to be run between client and server. For such applications, which should be covered by the general concept of the framework as well, the communication from the individual apps to the server must be channeled through a single Communicator, connected to the physical connection medium. Additionally, especially when looking at a control interface for all automated solutions using a touchscreen, one might want to switch between the different applications. Therefore, it is necessary to integrate the applications within a single executable that allows switching between the different GUIs.
This concept can be realized by using a multi-application Core. An illustration of this concept is shown in Figure 7.5. Using the developed and implemented abstraction layers, multiple APP-plugins can access a single connection object of the Core for physical interaction and message handling. Messages to be sent are pushed to the Communicator and then processed. As each APP-plugin provides its application identifier, messages can still be identified, processed, and filtered. For receiving messages, the server sends a message to the Communicator of the client computer if any of the running APP plugins associated with this Communicator has matching filter rules. The received message is then distributed to all APP plugins connected to the Communicator. Finally, filtering within each APP-plugin is performed in the same way as in the simple case with a single plugin.

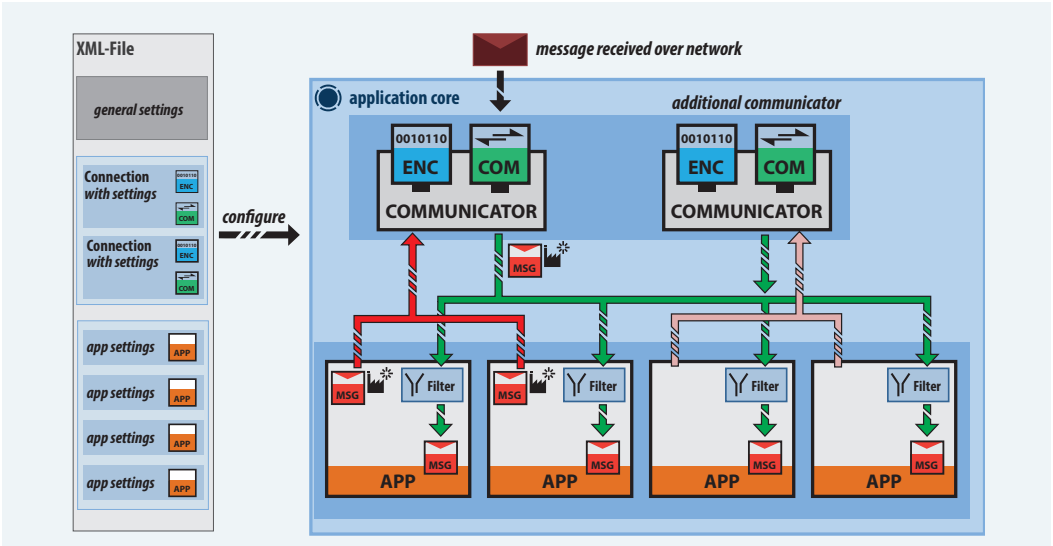


Figure 7.5: Illustration of the multi-application core, allowing multiple APP-plugins to utilize a single Communicator for message processing.

7.6 XML-based configuration

The configuration of the application and the used plugins, as described previously, can be performed entirely with a XML-based configuration file that is parsed and processed at the program start-up. This configuration includes many of the features already described, like the core functionality, the Connector definitions, and the used APP-plugins. This definition is separated into three parts: First, general settings regarding the application core are defined. This definition includes global constants like the path for plugins to be loaded and may be expanded by additional definitions.
The second part describes the configuration of the communication interfaces and Communicators using the definition of Connections. This includes the names of the required ENC- and COM-plugins with their settings. Such configurations include the selection of client or server mode, host names or IP addresses, and ports.
Finally, one or multiple definitions of individual plugins are included in the XML-based configuration file. After selection of the APP plugin by its name, the application identifiers and additional parameter are set.
Using such a configuration, the individual devices can be configured and integrated within the framework and adaptions can easily be performed. An example for such a configuration file is given in Figure 7.6.

Figure 7.6: Example for an XML-based configuration file for an application to connect to the developed 3M Bair Hugger interface as used within the implemented framework.

```
<?xml version="1.0" encoding="UTF-8" ?>
<control>
    <config>
        <pluginPath>plugins</pluginPath>
    </config>
    <connection id="con1">
        <com plugin="ComTcp">
            <mode>client</mode>
            <ip>comserver</ip>
            <port>5000</port>
        </com>
        <encode plugin="EncXml">
        </encode>
    </connection>
    <app plugin="AppBairHuggerConnector">
        <identifier>ICU1-BairHuggerConnector</identifier>
        <location>ICU-1</location>
        <com>con1</com>
        <comport>COM17</comport>
        <baudrate>9600</baudrate>
    </app>
</control>
```

7.7 Database

As mentioned in the motivation, a feature of the developed medical framework is the connection to a single, centralized database for storage of the messages with control commands and measurement results. The database is an important aspect for collecting the study data and retrospective analysis, but not required for the framework applications to run and perform automated therapy as the plugins should mainly rely on the framework's message structure and not on the information stored within the database. However, such stored information may be used to initialize a plugin during a restart or to obtain additional information not present in the current message stream.
Connection from applications to the database within the framework is realized by another APP-plugin. The filter rules for this plugin are set to receive any message. Therefore, the server will relay every transmitted information to the application running the database plugin. This APP has a connection to the database server and will process all messages for storage in an appropriate database table. For the performed implementation, the freely available and relational PostgreSQL database was used.
As the primary function of the database within the framework is data collection and logging of all events, its structure is optimized for fast writing performance, allowing to store all messages in a timely fashion. Data export then can be performed retrospectively, where no demand for a real-time application is required.

To allow easy differentiation between the individual patient identifiers, patient data is stored in individual schemata within the PostgreSQL database. The database structure is defined by a general header table, containing the information of the message itself and providing a unique header ID for additional information to be linked to. The database tables are created for data in the format of integers, real values, strings (less than 256 chars), and text. This is done to achieve an improved performance within the table structure as all entries within a table are of the same type. For waveform data from high-resolution sources with up to several hundred Hertz per second, an additional table is used. It stores the measurements, received as comma-separated data blocks, just like those to minimize database overhead.
The data content of all other messages is either stored in a raw data table as plain-text, or in individual tables for different data types, depending on the availability of a plugin to decode the message and the message type. Given the general message data format with key-value pairs, implemented for most messages, the appropriate tables are automatically selected for each key-value pair and information is stored and linked to the unique header ID of the message.
To maintain the order of information and avoid mismatching of individual messages in the time domain, the timestamp provided by the sender of the message is used. An additional timestamp is stored at the moment of writing the information to the database. This provides further safety against time offsets and may allow for retrospective restoration of the communication flow if time-stamps of one connected client are skewed, e.g. due to offsets introduced by daylight-saving time.

As an initial approach for data collection, message-specific database tables were used in the AP study for animal subjects ap004 to ap009. With a growing number of different messages and the performance drawbacks of tables with different types, this approach was replaced with the more general structure defined by tables for individual value types, as described prior, and used from animal subject ap010 onward. An overview and comparison of those two implemented and used database structures is shown in Figure 7.7.

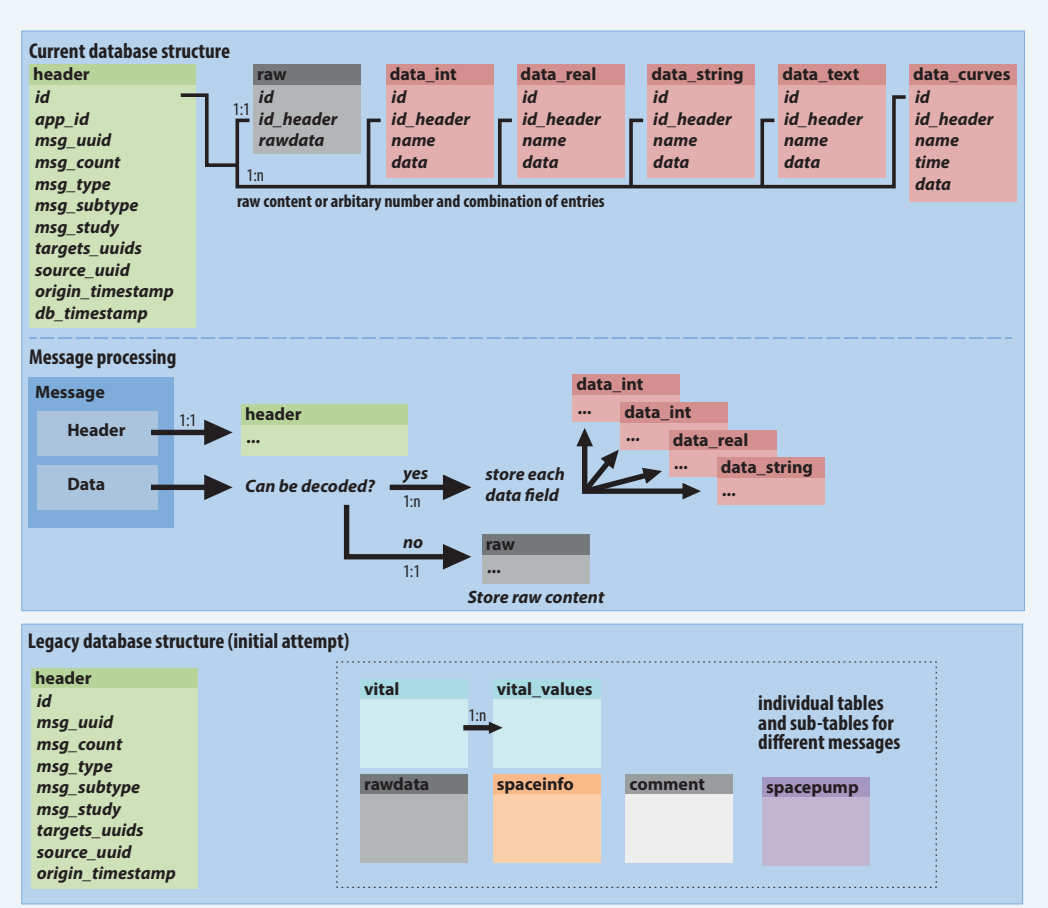


Figure 7.7: Message processing with the current structure (top) and the initial approach at the beginning of the AP study (bottom). In both approaches, an individual database schema is used for each animal subject and header information is stored in a separate table. In the current structure, tables for each data type are used and the information content of each message is stored in those type-specific tables, linked to the header ID (middle). In the initial approach, a separate table for each message type was implemented, containing the individual fields of the message with various data types. Within both approaches, a raw table is used to store unparsed message information if no appropriate MSG-plugin is available or the information cannot be parsed.

8 Implementation and Results

Using the theoretical and general concept described in the previous chapter, the framework was implemented as the Tübinger ICU Control and Monitoring System (TICoMS) for the specific needs in the experimental study settings at the Institute for Experimental Surgery at the University Hospital Tübingen. This chapter describes the specific implementation, some exemplary framework features, and the data export pipeline used for the retrospective analysis, which will be covered in a later parts of this thesis. Finally, the results of the performed studies are presented from a technical viewpoint in terms of collected data and data throughput as an evaluation of the framework performance.

8.1 Physical structure & implementation

Given the theoretical concept and software design, a first step was the establishment of a physical network between the individual medical devices. To provide an overview and as a reference for the achieved results, a short overview of the physical implementation is given in what follows.

8.1.1 Hardware setup & physical network structure

Implementation of a software framework for bidirectional communication of medical devices can only be performed upon an existing or to be established physical connection layer between the individual devices. Using the framework concept with modular plugins for message encoding and communication with a physical transport layer, few restrictions were given. The framework could be established using serial links, like industrial buses as RS232, RS485, CAN, Field/Modbus or other available technologies. However, for easy integration of computers, servers and available medical devices, a communication over Ethernet was chosen for the implementation of the framework.
The implemented physical setup mainly consists of two parts or locations: First, a bedside part for connection of all required medical devices and interaction with the system. Second, a backbone consisting of servers and storage for data processing and running central applications like communication server and database. As those parts are physically separated on different floors, network connections were established with two smart switches as central nodes. For load balancing and redundancy, link aggregation was used for interconnection. Several network services are provided on a running network server, including NTP, DHCP and DNS. The implemented setup is designed as a scalable solution for multiple beds. As many medical devices can be connected to a network and may interference with each other in the same broadcast domain, strict separation of signals on the physical layer for the network devices is required to avoid unwanted traffic and possible interaction of

the devices. This was achieved by implementing different Class B IP sub-nets. For central ICU infrastructure, the 172.16.100.0/24 sub-net is used, 172.16.110.0/24 for the first ICU setting, and 172.16.120.0/24 for a second one. The server room infrastructure and hardware management is encapsulated in yet another sub-net, 172.16.10.0/24. Of course, this is just a fundamental separation and classification and could be further scaled with additional sub-nets for different compartments and more sophisticated needs. Yet, the different sub-nets allow a clear separation of the traffic and routing between the individual components and units of the network. To avoid the usage of different physical network cables, VLANs for assignment of specific sub-nets to individual network outlets were used. This VLAN configuration for the individual network ports as well as routing between the individual sub-nets was performed with the used smart switches. A schematic overview of the used physical network setting and device interconnection is given in Figure 8.1.

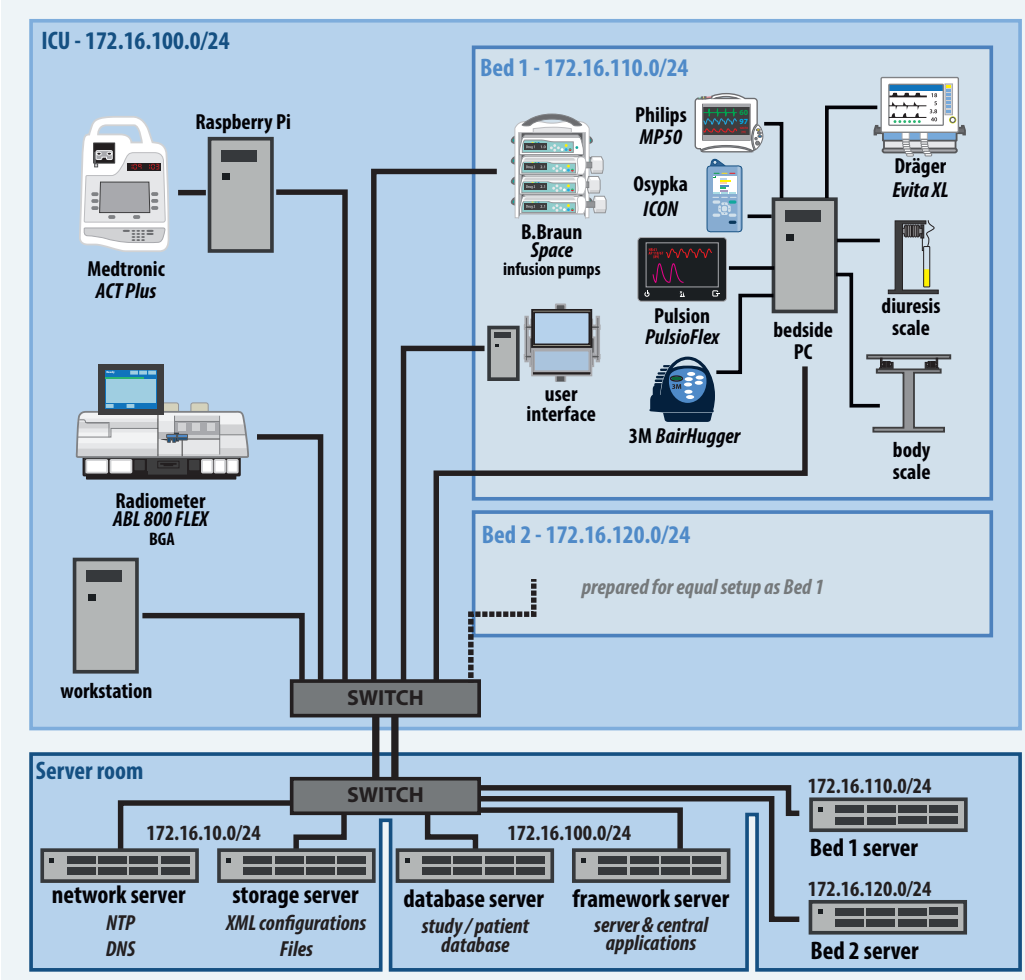


Figure 8.1: Illustration of the complete implemented physical structure of the entire TICoMS network. General classification consists of common infrastructure in the server room, common devices on the care unit, and individual bedside devices. Those individual compartments are divided into separated sub-nets to avoid cross-talk.

8.1.1.1 Central patient overview and control monitor

An ICU setting consists of a multitude of medical devices, each providing their own interfaces and screens. Moving towards an integrated form of intensive care, the increasing number of individual monitors and interfaces must be considered as is already far too high [3]. Given many interconnected medical devices with their information integrated into a common framework and the additional remote-control capabilities, a central interface is a next logical step. To provide this interface, a patient interface consisting of a monitor and a touchscreen was built and integrated in the framework. The implementation and schematic overview can be seen in Figure 8.2. The top monitor shows current and historic patient information, obtained from the vital parameters and settings stored in the central database. It provides a detailed overview of the patient with a user-definable observation window ranging from 15 minutes to up to 24 hours. The touchscreen at the bottom provides a general interface to a multitude of framework-based control applications.

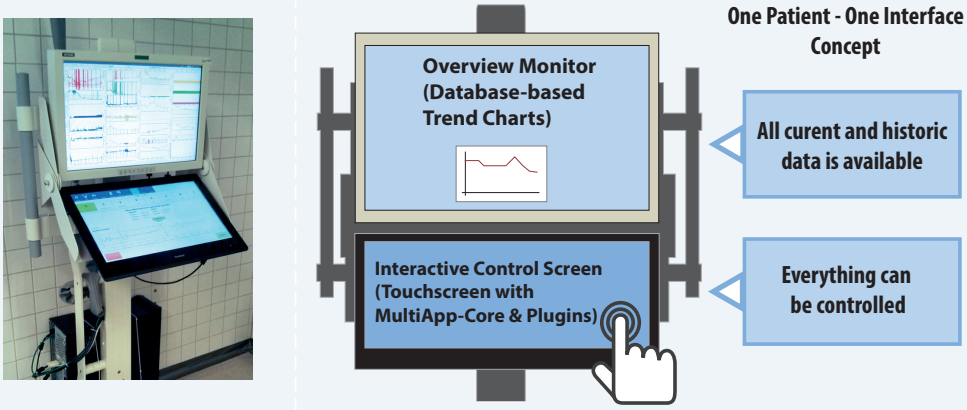


Figure 8.2: Patient interface with observation monitor on top of a touchscreen control interface. A Photo of the actual hardware (left) and an illustration (right), highlighting the central interface concept, is shown.

8.1.2 Applications and Plugins

Several plugins and applications were developed for practical implementation and evaluation of the framework at the Institute of Experimental Surgery at the University Hospital Tübingen. Mainly, two core executables are used: First, a single application core, used for standalone applications; Second, the multi-application core with a GUI interface, used in the central patient interface for selection of and interaction with various plugins.
For each of the used medical devices, a plugin for connecting the individual device to the message-based network was developed. The Dräger Evita XL ventilator and B.Braun SpaceStation infusion pump pillar, with SpaceCom network communication module, were integrated to allow for remote control and data collection. A Philips MP50 patient monitor was included to collect numerical values on a second by second basis as well as reading high-frequency curves up to several hundred Hertz. Weight measurements are collected from in-house developed diuresis and the body scales. A 3M BairHugger forced-air blanket

temperature-management unit was included via a developed interface board. Furthermore, laboratory measurements from a Radiometer BGA device and a POC ACT test device were added. For the latter one, a Raspberry Pi was used as the framework interface. Besides those device-connecting plugins, several additional and interactive plugins, mainly on the patient control monitor, were implemented. Those include manual control interfaces for the infusion pumps, ventilation device, temperature management unit, allow for paperless comments and manual inputs for parameters like weight, and implement interactive therapeutic protocols. Furthermore, plugins visualizing the overall system state and providing warnings, if devices should malfunction or not respond to the medical framework server anymore, were implemented.

8.1.3 Examples

To provide a brief example for the implementation of the proposed plugin architecture and message communication, the network communication and pump control is outlined and examples for the interactive touchscreen on the central patient monitor are shown.

8.1.3.1 Com-Plugins (TCP & RS232)

A central component of the message framework is the communication with other applications. As previously described, Ethernet connections are used for communication in the used experimental setting. A single COM-plugin is used for communication between all applications of the framework. The COM-plugin provides configurable options, e.g., for client and server mode, and delimiters for start and end of the message. Therefore, it is not only used for the communication between the framework applications, but utilized with an individually adapted configuration for the communication with the B.Braun Space Station, body scale, the Raspberry Pi for connecting the ACT device, BGA device, and other interfaces relying on an Ethernet connection.
For communication with other devices over serial port using the RS232 protocol, a second COM-plugin was developed. Equally to the TCP plugin, it provides an abstraction layer for communication between different devices by serialized message objects. For example, communicating with the ventilator, patient monitor, and diuresis scale.

8.1.3.2 Control of infusion pumps

The control, configuration, and reading of the infusion pumps of the B.Braun Space Station is performed by two important plugins. First, a device connector APP-plugin, performing communication with the pumps and providing base functionality like timers for bolus durations and processing of relative and absolute rate changes; Second, an interface for configuration and monitoring by the caregivers. An in-detail illustration of this setup is given in Figure 8.3 and described below.

Device connector The device connector is the control application that connects to the B.Braun SpaceStation via Ethernet interface. Therefore, a second Communicator is used by the application, handling the entire communication to the pumps. For communication with the framework, three types of messages are used: Two messages are used for information about the infusion pumps and sent by the application. First, a data message is used to inform

about the current infusion rates that are important for many applications. Second, a status message with information about the present pumps, their settings, and other information about the state of the infusion pump. A third message can be received by the application, allowing for pump control. This control message allows setting different parameters, including the pump rates, limits, minimal speed, bolus durations, and volumes. Given the desired input, the application plugin handles the timers for automated administering of bolus infusions and is a hard limiter for the user-defined infusion rate limits. Therefore, no application can override these limits by simply setting an infusion rate, but must explicitly adjust the limits. This is especially important for remote control applications with drugs like Arterenol, where even small infusion volumes and changes may have severe consequences.

Device interface The second application is the device interface on the patient overview and control monitor. This application allows the observation of the pump states and the control of the individual pumps, e.g., for setting the infusion limits and rates. It is only a GUI, showing the current state and sending pump control message based on the user input to the connector application. Yet, this application is a very important safety feature and plays an essential role as it provides the ability to override automated commands. Each pump control message has a flag to indicate if the message was generated automatically or triggered manually. Therefore, manual commands can be distinguished and override automated settings, e.g., if automated rate changes need to be reverted or an active timer-based bolus need to be stopped.

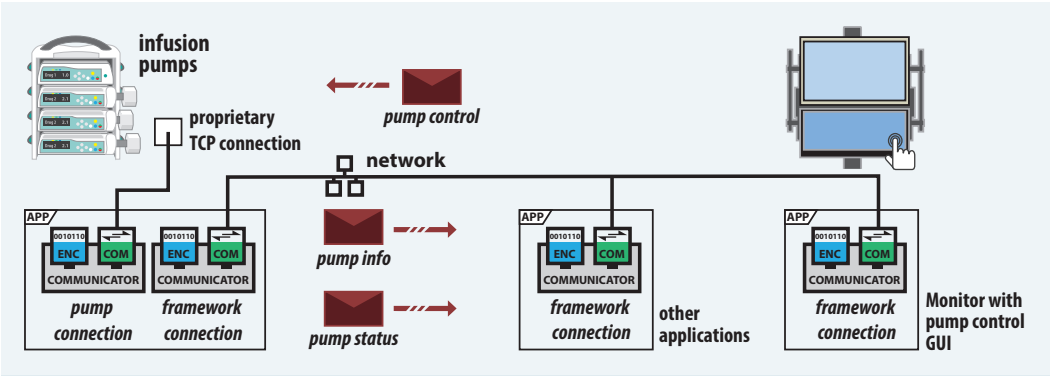


Figure 8.3: Overview of pump interaction with pump connector and GUI application. Status messages for the pumps are sent to the monitor and other applications. Pump control messages can be received from any sender within the network.

8.1.3.3 Interactive GUIs on the touchscreen interface

The developed central patient overview and control monitor is used for presentation of the current and historic patient data on the upper monitor and for interactive applications on the lower touchscreen interface. Using the developed multi-APP core, various framework applications can and controlled, including interactions with infusion pumps, ventilator and other medical devices. Selection of individual plugins can be performed with a tab-bar at the top of the interface. The GUI for the selected feature is then provided and handled by the appropriate plugin. As this interface is perfectly integrated within the framework

and can be easily expanded, it provides an essential component for automation of medical protocols and development of custom interfaces which could not easily be developed in other dedicated hardware. Examples for such plugins with touch-screen optimized GUIs are shown in Figure 8.4 for the control of infusion pumps and the interactive step-by-step study protocol for assessment of the volume state in the VNA study.

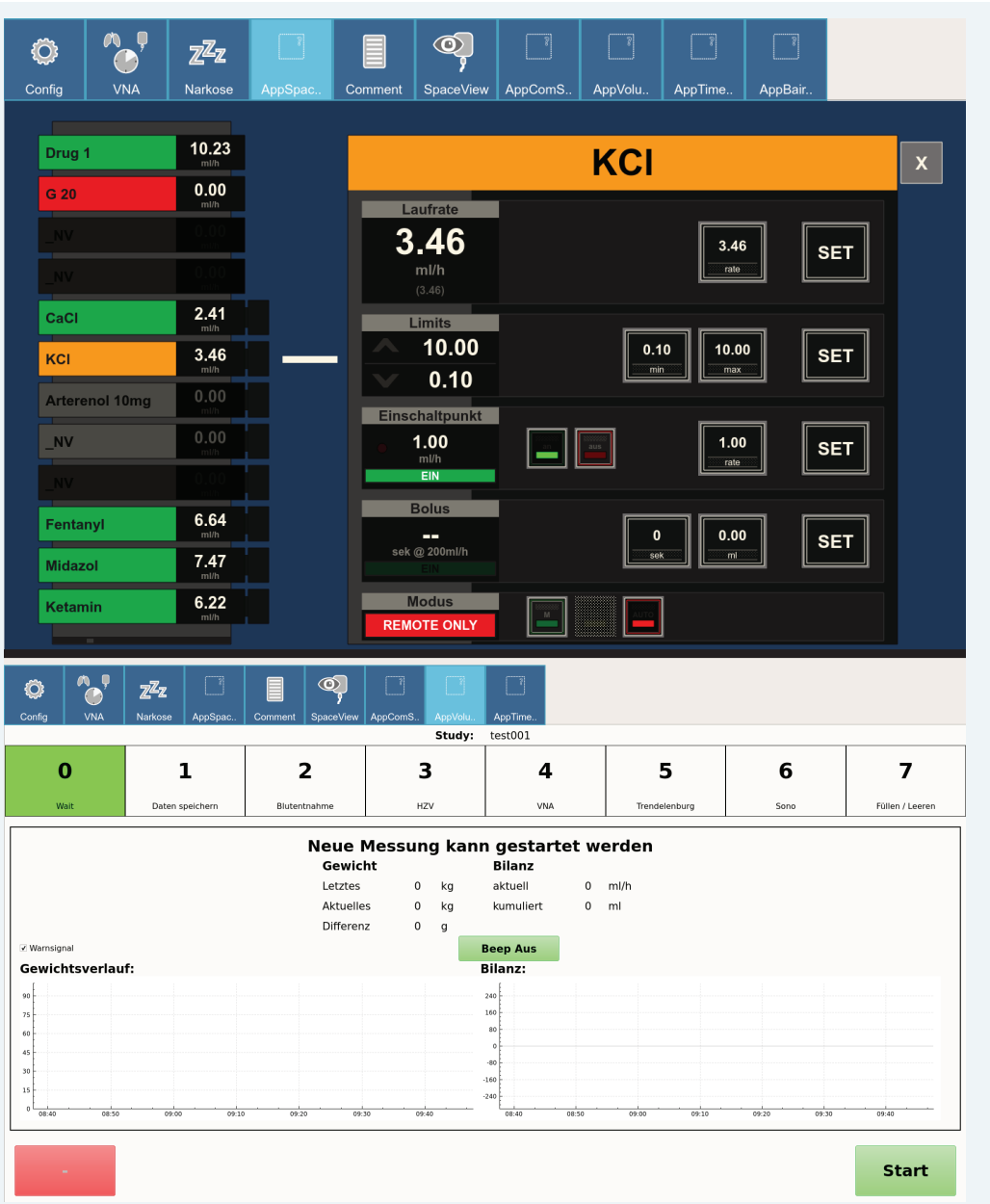


Figure 8.4: Examples for the GUIs of the touchscreen user interface. Pump-control screen for monitoring and manual control of the infusion pumps with colorized pump states including settings for rates and limits (top). Integrated step-by-step workflow for assessment of the volume state in the VNA study (bottom).

8.2 Data Export

Data export is an essential step for retrospective analysis and further processing of information with methods of machine learning. Given the stored information within the PostgreSQL database, export is performed with an R script pipeline. This entire data processing pipeline with all intermediate steps for export and documentation is shown in Figure 8.5. The generated results include overview plots for an animal subject as PDF files for documentation purposes with an overview of the vital parameters and the administered drug infusions. Further export processing of the individual parameters is dependent on the data frequency and separated in numerical data with frequencies up to 1 Hz and high-frequency wave data. Export using R scripts was performed for a variety of reasons, including the requirement that the documentation of the performed studies should be available all the times, which can easily be achieved with the simple PDF export by R and the fact that R is freely available software. Thus, in comparison to MATLAB no internet connection and license renewals are required, which would provide an obstacle for such an essential step of an isolated productive system.

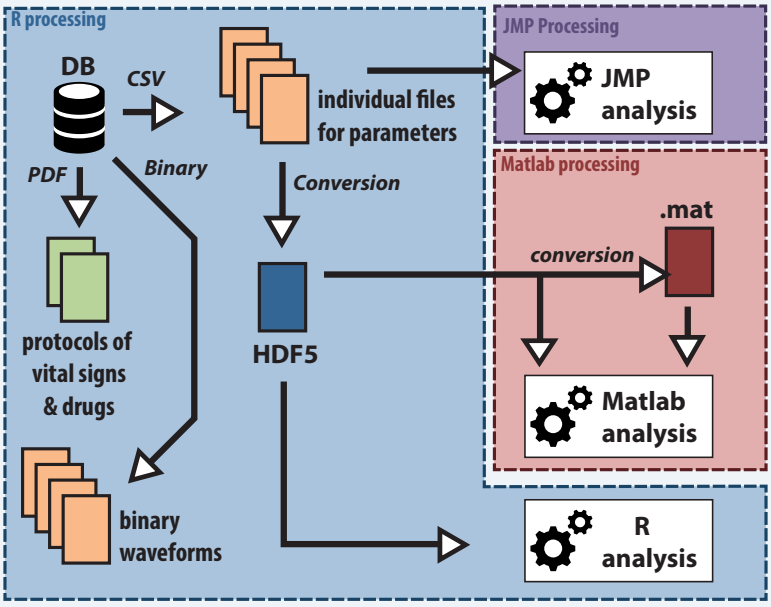


Figure 8.5: Data export pipeline for documentation and further processing of the obtained data in different file formats.

8.2.1 Numerical data

Numerical data is exported by an R script pipeline to individual CSV files for each parameter. This CSV files can be used for further processing with statistical tools like JMP or combined to other CSV files, containing a subset of parameters of interest. Additionally, the R scripts generate a HDF5 file as a compact storage solution for all numerical results. A schematic

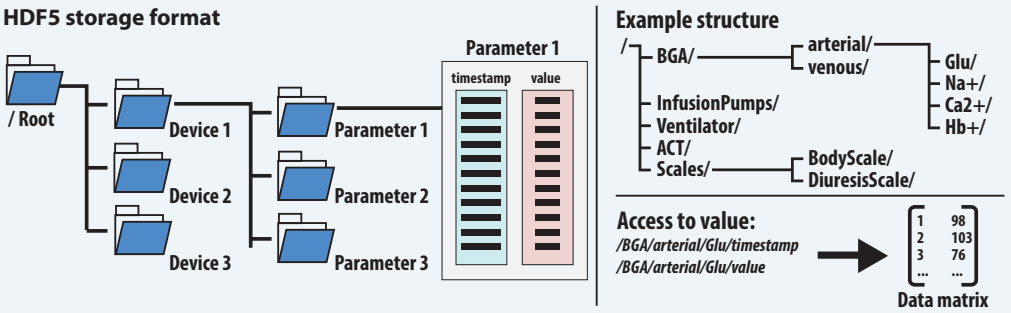


Figure 8.6: General HDF5 tree structure for storage of the results (left), example for structure within the used context (top right) and example for access to individual parameter by path names (bottom right).

overview of the export structure within the HDF5 format is given in Figure 8.6. As the HDF5 format provides a directory structure, each device is represented by its own folder. Each parameter of the device is then represented by an array of timestamps and values within this folder. Due to this directory structure, each parameter can be addressed by a unique path. This is especially useful, as the measured parameters may vary between individual trials. The generated HDF5 file can then for example be imported in MATLAB and used to create an intermediate .mat file for performance reasons. Then, further analysis, including the comparison of different animal subjects with machine learning may be performed.

8.2.2 Curve data

Besides numerical results, high-resolution curve and waveform data with up to 500 Hz is stored and exported for ventilation device and patient monitor. As this high-resolution data would yield enormous overhead by comma-separation of each measurement result, data export is performed with binary files. Within this format, timestamps and measurements are encoded by a fixed number of bytes without further delimiters. A short example for obtained wave data from the ECG-II curve of vna007 is shown in Figure 8.7.

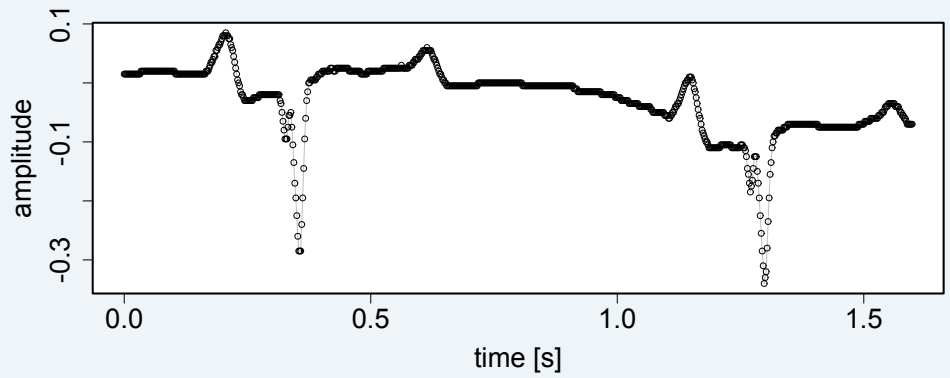


Figure 8.7: Short snippet of collected high-resolution ECG curve data from animal subject vna007.

8.3 Statistics & Results

The framework as described and implemented was used for measurement of all data during two performed and presented medical studies at the Institute of Experimental Surgery at the University Hospital Tübingen. However, besides any medical relevance, individual aspects of automation, and data analysis, which are described in later chapters within their appropriate context, the results of this full-scale implementation and long-term evaluation of the system with a significant amount of collected parameters and the achieved data throughput shall be presented from a technical perspective in the following.

8.3.1 Framework implementations

The implemented framework was expanded to 2 core applications, 42 APPs for connecting devices, user interfaces and automation controllers, an XML encoding plugin, 2 Communication plugins for Ethernet and serial communication, over 30 MSG plugins, and an UTIL plugin for HL7.
To give an estimate of the size of the developed and tested framework with various medical devices and interactions, a total number of 496 .cpp and 560 .h files with a count of 113 869 and 21 583 lines of code, respectively, and additional 24000 lines of comments for defining the interfaces and methods were implemented.

8.3.2 Integrated devices and protocols

In total, over 13 different data sources, including bedside and laboratory devices, were integrated into the developed framework for data collection and remote control. An overview of the different parameters, collection rate, and the animal subjects for which information was collected, is shown in Tables 8.1 and 8.2. For integration of the Philips patient monitor, the Philips IntelliVue protocol for medical devices is implemented. Communication to the Dräger Evita XL ventilator is performed using the Dräger MEDIBUS RS232 protocol. For integration of the Radiometer blood gas analyzer, the HL7 protocol was processed. More specific details regarding automation and further processing of the collected information will be provided in the subsequent parts of this thesis, covering closed-loop applications and data analysis.

Table 8.1: Overview of the collected parameters from the different devices and applications with temporal resolution and individual animal subjects for which those parameters have been collected.

Philips MP50 patient monitor	**Measurements:** CFI, external HR, dPmax, CCO, CCI, C.I. ABP_{dia}, ABP_{mean}, ABP_{sys}, CVP_{mean}, Perf, Pleth, T_{blood}, ITBVI, ITBV, SI, SVV, SV, EDVI, EDV, EVLWI, EVLW, Pulse	
	1 second	**ap004-ap010, vna001-vna010**
	Patient data: ID, name, BW, gender, age	
	event based	**ap004-ap010, vna001-vna010**
	Curves: 3 EEG channels, ABP, CVP	
	50-500 Hz	**vna001-vna010**
B.Braun Space infusion pumps	infusion rates, infusion names, positions within the pillar, pump configurations and settings	
	$\approx$ 2 seconds	**ap004-ap010, vna001-vna010**
Dräger Evita XL ventilator	**Measurements:** CO_2 production, Compliance, Deadspace, Endtidal CO_2, Expiratory Volume, Inspiratory O_2, Lung time constant, Mandatory Trigger, Mean airway pressure, Minimal airway pressure, Minute volume, PEEP airway pressure, Peak airway pressure, plateau airway pressure, rapid shallow breath index, relative deadspace, resistance, respiratory rate, spontaneous resp rate, spontaneous min volume	
	1 second	**ap004-ap010, vna001-vna010**
	Settings: ASB ramp, Apnea time, backup resp. rate, backup tidal volume, Flowtrigger, IEExp, IEInsp, $InspO_2$, Insp. pressure, Max. insp. pressure, Max. insp.pres. for CPAPASB, PEEP, RR, Tachypnea resp. rate, threshold of end of flowcycle, Ti, tube compensation, tube diameter	
	1 second	**ap004-ap010, vna001-vna010**
	Alarms: Alarm flags of the Evita ventilator device	
	1 second	**vna001-vna010**
	Curves: airway pressure, flow, insp. volume, start insp. cycle	
	50 Hz	**vna001-vna010**
Osypka ICON	raw data, age, ci-bsa, ci-wt, co, flc, gender, height, gr, icon, id, lvet, pep, si-bsa, si-wt, sqi, str, sv, sw, tfc, timestamp, vic, weight	
	1 second	**ap010, vna001-vna010**
Pulsion PulsioFlex	APdia, APsys, CPI, CPO, CVPmax, CVPmin, DO, DO2I, HP-PR, Height, MA, O2ER, PPV, time, SV, SVI, SVR, SVRI, SVV, ScvO2, Type cCO, VO2, VO2O, CCl, cCO, dPmx, meanCVP, alarm settings	
	1 second	**vna001-vna010**

Table 8.2: Overview of the collected parameters from the different devices and applications with temporal resolution and individual animal subjects for which those parameters have been collected. (cont.)

Radiometer BGA APL800 Flex	Type (Art., ven.), COHb, Ca^{2+}, Cl^{-}, Glu, HbF, K^{+}, Lac, MetHb, Na^{+}, O_2Hb, RHb, SBC, SBE, T, p50act, pCO_2, pCO_2T, pH, pHT, pO_2, pO_2T, sO_2, tBil, tHb, tO_2	
	event based	**ap001-ap010, vna001-vna010**
Medtronic ACT Plus	two individual time measurements, mean clotting time	
	event based	**vna001-vna010**
3M Bairhugger	temperature setting, speed setting, control commands	
	1 second	**vna005-vna010**
Diuresis scale	total volume collected	
	1 second	**ap004-ap010, vna001-vna010**
Body scale	netto weight, brutto weight, status bits (e.g., stable measurement), digital input pins (tilt detection)	
	1 second	**vna001-vna010**
Fluid balance	different derivative measures from the body and urine scale correlated to infusion rates	
	1 second	**vna001-vna010**
Volume Need Analysis	blood pressures at specific points within the maneuver, settings, and derivative values, sonographic results	
	event based	**vna001-vna010**
Comments	free text comments by caregivers	
	event based	**ap004-an010, vna001-vna010**

8.3.3 Collected data & runtimes

During the two performed studies, all data was processed and recorded by the database connected to the developed medical framework. In order to show the reliability and capabilities of the developed framework, the evaluated runtimes and the total number of stored database entries for all connected devices are presented. In total, data collection and evaluation of the framework were performed for 1.332 hours and resulted in a total number of over 47 million transmitted and stored messages with almost 8 billion database entries. A more detailed breakdown for each animal subject is given in Tables 8.3 and 8.4. Whereas the former table provides an overview of the runtimes and the total number of messages transmitted by the framework, the latter table presents the number of collected data entries in relation to the data format within the database for both used table layouts. As noted in the database description (7.7), the first format consists of tables for each message type, containing all values of the specific message. The enhanced format separates all information according to the message content in integer, real, string, text, and curve data for better writing perfor-

mance. Technical messages, like client-server interactions and other non-vital data were not processed further by the database and stored as raw strings. Variance and reduction in the number of the messages from the infusion pumps (B. Braun Space Station) can be observed for the old database format. This reduction of the pump polling frequency after animal subject *ap004* was required to maintain stable running condition for the entire study duration as supposed problems with the network buffer within the infusion pump pillar led to communication breakdown when using a higher polling frequency.

Table 8.3: Overview of the framework related statistics for the two performed studies.

	Animal subject	Collected param.	runtime [h]	Messages
AP Study	ap001	_(1)	_(1)	_ (1)
	ap002	_(1)	_(1)	_ (1)
	ap003	_(1)	_(1)	_ (1)
	ap004	171	65	631,192
	ap005	248	72	862,207
	ap006	250	36	433,697
	ap007	245	72	1,017,164
	ap008	253	52	1,648,579
	ap009	246	68	999,880
	ap010	229	13	177,342
VNA Study	vna001	331	96	4,261,622
	vna002	321	57	2,137,001
	vna003	329	96	4,366,997
	vna004	347	96	4,458,760
	vna005	350	96	4,668,021
	vna006	349	96	4,844,387
	vna007	349	96	4,712,003
	vna008	348	96	4,821,337
	vna009	350	96	4,800,420
	vna010	397	48	2,381,399
	total		1332	47,222,008

(1) For animal subjects ap001–ap003 data was not yet collected in a central database. These animal subjects of the ongoing clinical studies were used for collection of first real device data and implementation of the device connections.

8.3.4 Network monitoring and statistics

To maintain an overview of the entire system, a Paessler PRTG [192] network monitoring system was used. It provided the essential tools for monitoring the running conditions of all connected devices prior and during the performed studies and for detection potential problems and failures. As the system collects network statistics of all included devices, these log files can be used for a brief evaluation of network performance and individual device and server load while running framework applications during the conducted studies. An overview of performance statistics is given in Table 8.5 to show the average data rates handled by the individual components of the framework. Obviously, the highest load can be observed at the server, handling all messages with an average data rate of 60.48 MB/s.

Table 8.4: Overview of the number of individual database entries collected during the two performed studies with old (top) and improved (bottom) database format as shown in Figure 7.7. In the old format, entries were separated by message type, in the new format by the data type.

subj.	spacepump	spaceinfo	vital	vital_value	cmt	raw
ap001	_(1)	_(1)	_(1)	_(1)	_(1)	_(1)
ap002	_(1)	_(1)	_(1)	_(1)	_(1)	_(1)
ap003	_(1)	_(1)	_(1)	_(1)	_(1)	_(1)
ap004	3,643,872	331,273	229,342	4,806,954	0	577
ap005	1,278,201	106,524	751,717	16,478,432	47	3919
ap006	643,862	55,917	375,604	7,761,266	28	2146
ap007	1,284,829	107,196	800,470	16,317,540	25	109,473
ap008	2,054,269	171,329	1,302,250	28,092,128	37	174,963
ap009	1,207,051	100,595	796,567	16,873,684	24	102,690
total	10,112,084	872,834	4,255,950	90,330,004	161	393,768

subj.	int	real	string	curve	text	raw
ap010	2,813,818	3,523,466	5,138,301	21,328	0	17,625
vna001	23,203,728	36,282,600	38,262,653	696,121,883	0	113,394
vna002	13,942,500	19,589,726	23,290,228	416,777,581	0	69,613
vna003	23,751,801	35,065,238	39,172,263	759,641,273	0	118,736
vna004	24,353,045	36,517,198	39,630,009	772,456,394	0	121,271
vna005	27,537,384	35,517,571	39,226,083	750,241,651	0	119,129
vna006	27,003,387	37,550,681	38,586,663	763,503,660	0	118,687
vna007	26,656,903	35,265,170	39,209,214	758,786,855	1	118,006
vna008	27,895,840	36,982,688	39,599,740	772,611,988	0	119,870
vna009	27,107,424	37,385,694	39,834,473	764,798,344	0	117,196
vna010	13,780,297	17,468,900	20,450,167	396,455,021	0	62,873
total	238,046,127	331,148,932	362,399,794	6,851,415,978	1	1,096,400

(1) For animal subjects ap001–ap003 data was not yet collected in a central database.

Table 8.5: Median data rates for the individual network devices during the VNA trials. Trunk is the backbone network connection between the ICU and server room switch, Server is the load of the network connection to the server, running the application server, database, and network services. The embedded PC is used for collection of measurements from patient monitors, scales, and 3M Bair Hugger.

Network connection	$\widetilde{in}$ [Mbit/s]	$\widetilde{out}$ [Mbit/s]
ICU embedded PC (data collection)	13.41	91.41
ICU SpaceStation (infusions)	2.13	9.68
ICU Monitor (User Interface)	126.61	25.10
Trunk (Switch interconnect)	161.31	461.05
Server (Server with VMs)	483.86	177.02

9 Discussion

When dealing with medical devices, one of the most important problems is the lack of standardization and interoperability [7, 10, 11]. Thus, any further processing or analysis based on data gathered from such a loose collection of individual medical devices is hardly possible until this problem is solved. Given the rare opportunity for practical evaluation of possible solutions for this problem in an experimental porcine ICU setting with less restrictive legal boundaries than in clinical practice involving humans, a novel, modular framework was developed. This system allows for interconnection of medical devices using message-based communication in a client-server environment.
The presented framework was especially designed for experimental academic research with the main focus placed on scalability for fast and reliable expansion of the system regarding changing and evolving research questions. Besides this aspect of fast expansion, the re-usability of implemented parts was important during development as it allowed for faster testing and debugging of established components. Thus, components required in any or almost all applications, like communication plugins, were reused and potential flaws and errors were successfully discovered and resolved faster during the expansion and development phase.
An important aspect of the framework is the modular, plugin-based concept of information exchange using individual message objects that are dynamically loaded from specific plugins. Using this approach, only the general message structure but not the specific implementation must be known to all clients beforehand. Only devices interested in the message content require the respective plugin for further processing. This allows standardized message-based communication between the individual medical devices without the need for alternating the existing applications and implementation. If a device receives a message, the header is processed and if no message plugin for decoding is available, the message simply is discarded. Therefore, additional messages can easily be included, maintaining a backwards comparability to the already implemented system. This is an important feature, as in the case a message arrives at a wrong target by accident, this must not result in any invalid states or problems. As another step to ensure that only messages intended for the receiver are further processed, message filtering after successful parsing is performed. This utilizes a set of rules and identifiers, allowing arbitrary combinations and exclusions as each filter can individually be designed for the appropriate need.

In comparison to various independent solutions for measurement and control of single parameters as presented in Chapter 3, the usage of a software framework with a central database provides advantages in terms of data synchronization and collection. An ensemble of individual data sets, collected on different devices, for example various laptops, can hardly be processed any further and matching of the timestamps may be very difficult.

Additionally, many devices with large storage space are required for collecting all data, rendering such a solution not scalable. Solving this problem, frameworks as for example proposed by Feng et al. [111] are capable of collecting data for retrospective analysis. However, such solutions still lack the ability for connected clients to communicate with each other or for a server to send and relay control commands to the clients. Yet, this is a requirement for anything more than retrospective data analysis and only achievable with full-scale communication solutions like the developed framework.
By using a client-server based system, the server has the control authority over the connected devices and information exchange. Additionally, it can provide an overview of all connected devices. As all information is forwarded by this central node, collection of all data and control commands can easily be performed and stored within a connected database. This allows for the retrospective analysis of all performed actions. Besides advantages during development and debugging, such a feature may be important for automation within the clinical setting. If any malfunction or patient harm occurs, a complete track of actions can be obtained and investigated. Due to the general message format, each message can be logged at least as plain-text within the database. If messages are specifically encrypted, e.g., for proprietary communication between devices of a single manufacturer, such logged messages may retrospectively be decrypted to gain a detailed insight of the performed measures.
Of course, such a client-server approach also has its drawbacks. Its central interaction point provides a single point of failure as the large number of messages has to be handled by the server to avoid clogging the network and cause severe delays. Regarding this weak spot and for better scalability, peer-to-peer approaches as for example suggested by Pölsen et al. [110] can be beneficial. Yet, such approaches face other difficulties, like maintaining an overview of the network and provide access control. Furthermore, problems regarding a single server might be countered by implementing redundancy with multiple instances, sharing the network load and providing a fail-safe switchover. Thus, each solution is a tradeoff between the various goals to be achieved. Yet, regarding the research scope for academic purposes and early prototypes, the client-server concept allowed a better overview and control of all devices and messages and thus was the concept of choice.
Software concepts with many possible and theoretical aspects are a first essential step, but practical complications are easily overlooked. Thus, the experimental evaluation of such as system was an essential step. For example, problems like connection loss must be addressed in any medical framework. This includes questions like intermediate storage and automatic retransmission of messages and notification of caregivers. Additionally, strategies for returning to an operating condition from such a state need to be considered. One example is a short outage of the connection to the server. Once this connection is re-established, all clients would try to reconnect to the server at once. Obviously, such a load peak may cause further problems. Thus, randomized reconnect times for reduction of such events were implemented.
Furthermore, questions regarding the physical implementation and hardware have shown to be at least as important as design and implementation aspects and must not be forgotten in any further or related research. Given the heterogeneous landscape of numerous medical devices, a large number of connections needs to be established for interconnection. Even

in the used small research environment, a significant number of devices and servers is already present, requiring reliable and structured network and server infrastructure to handle the data. As often many similar devices like infusion pumps or patient monitors are used, cross-talk and interferences between those devices needs to be considered as messages of the correct format might be obtained by the wrong receiver. Whereas hopefully further checks ensure that no wrong commands are processed, filtering and processing may lead to a significant overhead and problems, especially if a huge number of devices would be interconnected within a single hospital-wide network. Therefore, separation of the IP network into different sub-nets and broadcast domains should be performed. For the developed system this was implemented by using separate VLANs, working independently on the same physical network connection. VLAN management and routing between the individual nets was performed with smart switches, which allowed for easy set-up and adaption of the sub-nets and VLAN, assigned to individual network ports. Network infrastructure also provides other important features like link aggregation and STP for finding the shortest path and avoiding network loops. Caused by manual mistakes like erroneously connected network ports, such events may take out the entire control systems with potential patient harm.
For easy configuration of the application's connections to the server and medical devices by unique device names within the XML-based configuration file, identification was performed using the DNS for forward and backward resolution of the IP addresses. Fixed IP-address were used and automatically assigned to the clients using a DHCP server whenever possible. Another important aspect is the management and synchronization of the real-time clocks of the individual devices. As all data should be collected and matched in the time domain for further analysis and processing, a common time base among all connected devices is required. This aspect was considered by using the NTP for time synchronization, providing consistent timestamps among all devices and messages. Besides that, maintaining an overview of the network and all connected devices was an important aspect to assess the operating condition of the system as a whole. Whereas each device could be checked individually, network monitoring was used to maintain a tight control and overview of any connected device regarding their network connection and other aspects like remaining storage space. This allowed a fast and complete assessment with notifications of potential failures at any time during the performed studies. Especially, considering large scale implementations of such medical frameworks, network monitoring will play an increasingly important role as it would not be feasible to observe each device individually.

Using the implemented framework, over 1300 hours of observations for over 300 parameters and settings could reliably be processed and stored in a central database. As this included all collectable parameters and settings provided by the medical devices, the maximum amount of information was preserved and is available for detailed retrospective analysis. Thus, minimizing the potential need for additional animal studies if specific details or correlations between parameters of interest or device configurations were not initially considered but become relevant during further analysis. Yet, despite the large number of connected devices, still not all information can be stored. Unfortunately, many

devices only provide data output on a second by second basis, whereas internal calculations are performed with a much higher resolution. Therefore, high-resolution data exceeding 1 Hz could only be stored in the central database for a subset of parameters from ventilator and patient monitor.

Whereas communication between different devices of the framework is based on transmitted messages, the database is only used for retrieval of historic data and retrospective analysis. Thus, the most important aspect was storing the collected information in a rapid fashion with optimized write operations. To facilitate this aspect, the data obtained from messages is separated into individual database tables of specific data types. For extraction of the information and further processing, matching of all obtained information in the time domain is one of the most important aspects. Using the central database, this can easily be performed as the timestamps obtained from the devices, synchronized by NTP, and the current time at the moment of writing into the database are stored for each entry. This second timestamp is especially useful if any communication delays have occurred or for example daylight-saving time settings of the connected devices cause offsets.
As information from multiple patients or animal subjects needs to be stored simultaneously, an individual database schema is created for each subject. This allows easy querying of the stored information for export and allows subjects to be easily removed from the database as well. Especially with a large number of patients or animal subjects in continuous usage, this separation helps to manage and clean up the stored information.
Data export from this central database is currently performed by a broad variety of ways depending on specific research questions and the tools for further processing. Furthermore, given changing experimental settings with different device configurations, infusions, and obtainable parameters, changing datasets between each animal subject are the norm. Therefore, export of a result matrix with fixed parameters was not feasible as further processing would need to keep track of the location and availability of each observation variable. Another aspect regarding the data export was the large amount of data with often repeating measurements, especially for device settings. A solution for those problems and challenges was provided by using the HDF5 format, supporting file compression for the repetitive numerical observations, resulting in a single export file of around 80 MB for all parameters of a single subject with 96 hours of observation. Additionally, utilizing a folder-like tree-structure, the HDF5 format allows to access the information by path names like `bga/arterial/glu`. This removes the need for numerical parameter IDs with the potential for offset errors in variable parameter sets. Despite all these advantages, many features of the HDF5 format are only available to a limited extend in the used R and MATLAB packages. Utilizing both tools, only the intersection of the respectively supported features could be used. Given the full potential, HDF5 may for example allow improved compression and support tables containing different data types, thus providing a more compact data structure. For high-frequency data of ventilator and patient monitor, binary encoding was used to save space, resulting in around 2 GB for each parameter of an animal subject. As this is a significantly larger amount of data, additional storage and processing capabilities, and adapted handling for further processing and analysis may be required. However, since high-frequency data has only been collected for the most recent

animal subjects of the VNA study, no further processing with the collected information is yet performed. Such next steps may include analysis of the pressure curves and ECG using wavelets or other analysis methods in the time and frequency domain [132–134]. Additionally, given the amount of collected data and the different methods for data export, the proposed data collection and export is a first and fundamental step for further analysis and already the foundation for several medical doctorate theses.

Beside the potential for retrospective analysis, the collected data was used to observe and facilitate therapy by the caregivers. Given over 300 collected parameters, aggregation and presentation was another essential step to maintain an overview of the animal subject and allow for easy interaction with automated systems. Especially with an increasing degree of automation and more medical devices in intensive care settings, central interfaces, combining the many various information sources, are required. For usage with the proposed framework, a monitor and a touchscreen interface were included as the central patient overview and control system. This allowed the implementation of various GUIs for framework-based applications, presenting current and historic patient conditions, allowing the control of medical devices and automation settings, and interactive and semi-automated processing of stepwise study protocols. Yet, the most important aspect was the ability for manual intervention to the automated control systems in case of potential critical situations and erroneous actions. Lastly, apart from those various use cases of such a touchscreen and the easy interaction, the glass surface provided an optimal solution regarding hygiene and cleaning concerns in the experimental setting.

9.1 Outlook

The framework was successfully implemented and evaluated during the performed studies. Yet, several features may be added in the future. Developments regarding standards for medical devices, like IEEE 11073 [114], are to be observed as those might guide further implementations or redesigns of the framework and may help to facilitate better connection to novel medical devices and their integration in the existing environment.
Given the academic focus, any clinical application would require several critical and important steps like freezing the current state, considerations of several additional safety measures, and formal verification. Even though safety-critical aspects such as data encryption, checksums, and verifications were considered during development, they still would need be added to the communication and encoding plugins in a further locked down design iteration, potentially requiring a re-design of some safety-critical aspects. Yet, given the research focus, a spin-off of individual features for clinical application seams more feasible than converting and locking the existing framework for such an application.
Based on the current implementation, better methods for configuration of the applications and settings may be implemented as configurations are currently defined at program start but not adaptable during runtime. For long-term application, dynamic adaption of running applications or even remote configuration through the communication network would be beneficial. Another aspect, especially beneficial during development, would be an auto-update feature for the individual applications. Once a new version of a plugin is compiled, it currently must be copied manually to each target. During compilation of each plugin, a

version number and timestamp are already added. On program start, each program core could compare the copy of the plugins stored in the local plugin folder with versions in a remote shared folder or query a version database. Once a new version is detected, it could be downloaded. This would be especially beneficial for plugins shared by many applications like the COM-plugin and reduce the potential error risk by continuing to use an obsolete version.

As the framework was developed to be platform independent, applications could be made available for a broad variety of operating systems and devices. Currently, implementations are only available for 32- and 64-bit Linux and Windows platforms. Initial tests were performed for ARM architectures of android devices. Such implementations would allow for a mobile control and monitoring interface, e.g., based on an Android tablet.

Further improvements could be made regarding a more advanced replay system, allowing stored information to be converted to messages and resent for distribution through the framework. Whereas individual messages already can be resent and simulator plugins for many devices have been developed to inject artificial messages and measurements for debugging purposes, such a general playback system would be useful for evaluation of new features.

As discussed earlier, the server is single point of failure and currently the system is dependent on a network connection between the central server and each device. However, beds and device groups may be moved. This may result in short, intermediate communication loss. Especially automated systems, like closed-loop controls should continue patient management during these times. Therefore, a compartmentalization and encapsulation of the features for a more autonomous bedside system should be performed. A local bedside server should manage all devices for a single patient and may be moved with them. Additionally, temporary storage of the collected information should be performed. Once again connected to a clinical network, this bedside unit should re-synchronize with the central database and server. An illustration of such a cascaded concept of bedside server and closed-loops within a central infrastructure is shown in Figure 9.1 This concept may even be further extended by the introduction of intermediate servers for individual hospital wards or units, thus limiting the frequent information exchange to smaller units and only communicating with the central infrastructure at a reduced rate for long term storage, management, and billing aspects. Alternatively, such problems might be partly countered by using wireless transmissions. However, given the large amount of data to be transmitted and still many scenarios with limited connection, e.g., elevators, those aspects cannot be omitted.

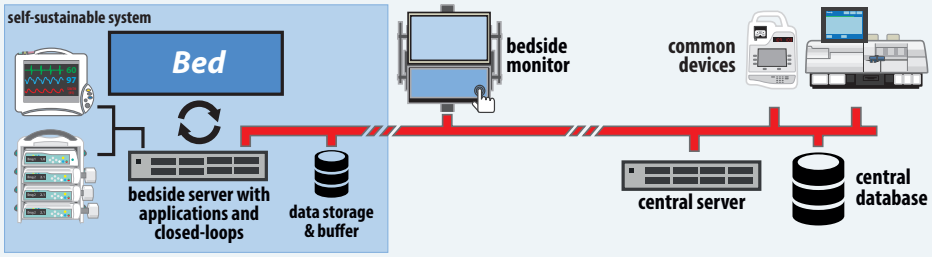


Figure 9.1: Concept of a self-sustainable bedside system with a local server for uninterrupted closed-loops and local storage to allow for transport to another hospital unit where reconnection and synchronization with the central system may be performed.

Part IV

Framework-based closed-loop applications

10 Motivation

Within the given medical research context of the conducted studies with porcine animal subjects, the developed software framework with its data collection and device control capabilities, as describe in the previous part of this thesis, was used as the foundation for the development and evaluation of several closed-loop applications.
During the ongoing studies, the main goal was collecting clinical research data to obtain a reliable dataset for further analysis within the medical context. To achieve this goal, well defined study conditions had to be established and maintained. Additionally, given the medical protocols to be carried out during those studies, repetitive manual steps were subject to automation to reduce workload and human error potential.
The most important factors for achieving those goals during the performed clinical studies were the successful management of hemo- and homeostasis in the individual animal subjects. To maintain hemo- and homeostasis, automated closed-loop management was applied to the three most relevant aspects within the given study context: The management of the body temperature, management of anticoagulation and management of blood gases, electrolytes, and glucose as the sole source of nourishment during the observation period. These three aspects of automated management will be covered in the following chapters, each including an individual clinical motivation, methods, results, and a discussion within the context of the performed clinical studies:
Anticoagulation management, presented in Chapter 11, was important for maintaining hemostasis and reduction of workload during the performed studies as frequent ACT measurements and adaptions to the heparin infusion pump had to be performed to avoid thrombosis or internal bleeding during the prolonged bedding times of the animal subjects.
Regarding the body temperature, presented in Chapter 12, the main goal was to stabilize the temperature and avoid hypo- as well as hyperthermia as such changes can drastically interfere with various biochemical processes and lead to physiological changes influencing homeostasis.
Finally, the most important aspect was the management of the electrolytes, blood gases, and glucose using fuzzy-logic based closed-loop systems, presented in Chapter 13. By controlling the RR, the gas exchange and end-tidal CO_2 ($etCO_2$) can been managed, thus stabilizing the blood gas concentrations and improving homeostasis. Furthermore, closed-loop systems for managements of individual electrolytes and glucose were implemented by utilizing fuzzy logic controllers to adapt to the specific needs of each individual animal subject during the course of the study.

11 Anticoagulation Management

11.1 Motivation

As the porcine animal subjects in the conducted studies are observed for up to 96 hours, therapeutic anticoagulation management to prohibit thrombosis due to the long bedding times had to be performed. This task has been carried out manually for the AP study using a Medtronic ACT Plus device for measurements of ACT and manual adaption of a heparin infusion pump. Due to the many required manual steps, such a task has a high workload and risk for human errors. Within this context, automated management of blood coagulation therefore was a prime example and use case for every day procedures in intensive care. Using the developed framework, various parameters and conditions can be observed and appropriate tasks can be automated. However, faced with the task of automating ACT processing, obtaining the measured clotting times provided a challenge. Unfortunately, currently no point-of-care device, allowing a direct event-based export of individual measurement results, is available and existing solutions are limited to retrospective export using serial ports or even still floppy disks. To bridge this gap, an automated solution was developed and published [193]. The following part recites the development process and the evaluation in depth.

11.1.1 Workflow

The manual task of anticoagulation management is prone to errors, especially in times of high workload, as several steps need to be performed. The general workflow is illustrated in Figure 11.1. After performing the measurement in a first step, the resulting clotting times and their average have to be observed and noted. Next, the new heparin rate has to be calculated manually according to a therapeutic look-up table like the one shown in Table 11.1. This calculation involves consideration of patient's body weight and the heparin dilution, which may need to be looked up. Finally, the now calculated rate has to be manually entered and set at the corresponding infusion pump [193].

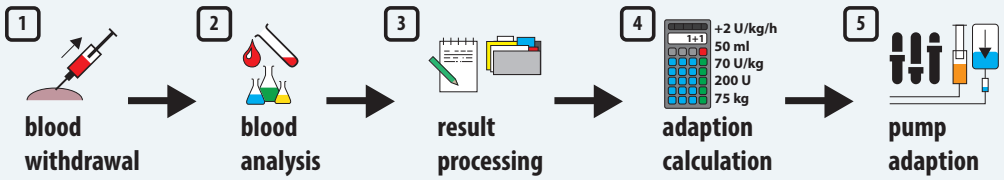


Figure 11.1: Workflow for manual ACT measurement, consisting of several steps including manual calculations and adaption of the infusion pump.

Table 11.1: Therapeutic heparinization chart based on [194] with a reduced bolus. Measured mean ACT is used for deciding the correct action and calculating the new continuous and bolus rates dependent on patient weight and heparin dilution [193].

Measured ACT [s]			Action	Rate adaption
Initial			Bolus 70 U/kg	18 U/kg/h
	<	70	Bolus 70 U/kg	↑ 4 U/kg/h
70	–	90	Bolus 35 U/kg	↑ 2 U/kg/h
91	–	110	none	no change
111	–	130	none	↓ 2 U/kg/h
	>	130	Pause 1 h	↓ 4 U/kg/h

11.2 Method

Besides the tasks of blood sample collection and performing the actual measurement of the clotting time, which still must be performed manually, processing of the measurement results, calculating the rate change, and applying it to the bedside infusion pump are tasks that can be automated. Due to the lack of appropriate interfaces for data export, the biggest challenge was obtaining the measurement information. Therefore, possible new export options like measuring electrical signals or visual recognition had to be considered. The most promising and simple approach was accessing the 7-segment displays of the Medtronic ACT Plus measurement device as they provide an easily visible and accessible representation of the required information. In comparison, the results displayed on the LCD are very small, have a narrow viewing angle, and are more complex to export electrically or using image processing. As first tests and analysis of the device showed easy electrical access to the 7-segment displays, this option was chosen and implemented.

11.2.1 Hardware Development

The device was dissembled and the printed circuit board (PCB) with the traces to the display was analyzed. All relevant electrical signals to the display are transmitted through a 40-pin ribbon cable, which was replaced with another cable with three pole plugs to branch of the required information. Observation of all signals was performed with an oscilloscope and the corresponding data lines and frequencies were identified. The next step involved the decoding of the multiplexed signal of the two 7-Segment displays. A Raspberry Pi [195] Version 1 Model B+ with a 32-bit ARMv6 processor was chosen as an appropriate device, as is provides a sufficient amount of input pins and an Ethernet connector for integration in the medial framework. As the electrical signals are branched off after the current limiting resistors for the LED segments, difference in logical voltage levels compared to the Raspberry Pi general purpose input/output (GPIO) pins are present. Thus, an adapter board for connection to the Raspberry Pi has been developed by using open-collector differential comparators for the individual signal lines and pull-up resistors for logic level shift. Schematic conversion and developed board are shown in Figure 11.2, the entire hardware

setup for ACT measurements including the adapter board and Raspberry Pi is presented in Figure 11.3.

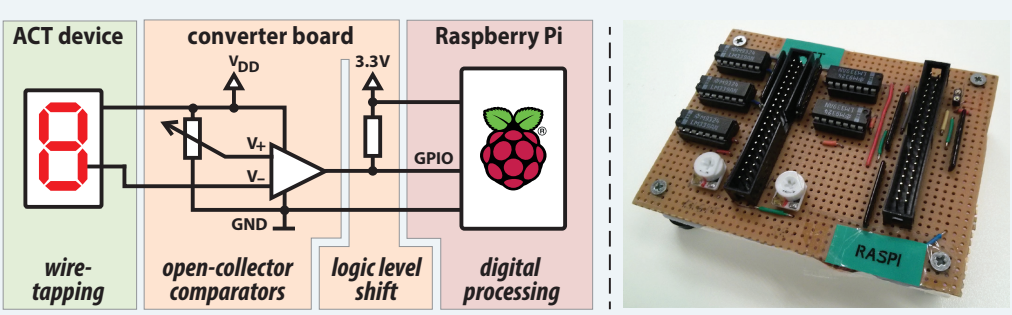


Figure 11.2: Schematic conversion flow for an individual digit (left) and the hardware board (right) as an interface between Medtronic ACT Plus and a Raspberry Pi.

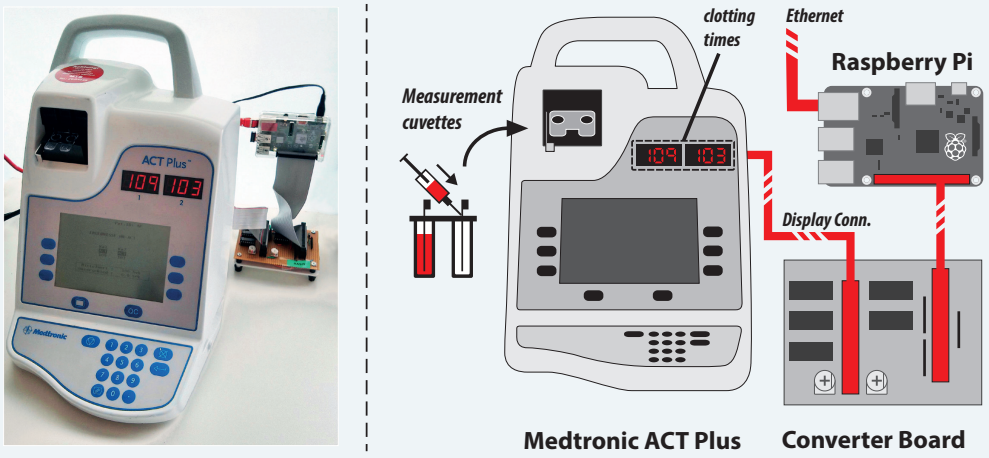


Figure 11.3: Photo of the ACT measurement setup with Medtronic ACT Plus device, converter board and Raspberry Pi for connection to the TICoMS framework (left, [193]) and schematic illustration (right).

11.2.2 Software development

Signal processing and analysis of the converted signals is performed on the Raspberry Pi using C++ and the Wiring Pi library [196] for accessing the GPIO pins connected to the segments via the developed converter board. Using the information of the lit individual displays, demultiplexing of the digits is performed, resulting in a 3-digit number for each of the both displays. For enhancement of number recognition and error reduction, consecutive readings are collected and the most often occurring number is selected by majority vote. This process is illustrated in Figure 11.4. For automated detection of start and end of a measurement, a state machine detecting zero times and invariability of the recognized numbers for a defined short timeframe is used. After detection of the measurement end,

the mean value is calculated and, together with the two raw readings, send to TICoMS. The message is then processed by an application, implementing the therapeutic table from Figure 11.1, calculating the infusion rates with patient weight provided by the user through a GUI, and sending the new rates to the infusion pump. This workflow is illustrated in Figure 11.4.

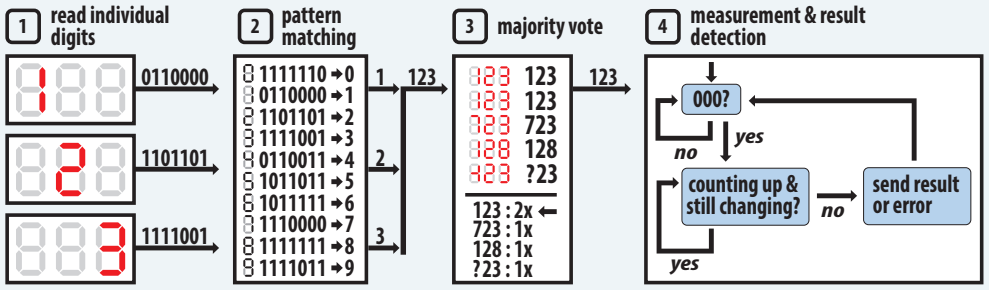


Figure 11.4: Automated ACT measurement workflow with pattern recognition. Individual digits are processed to obtain a numeric representation by matching the bit patterns. Detection errors are reduced by aggregating several readings and selection by majority vote. Then a state machine is used for detection of start and end of ACT measurements.

11.2.3 Evaluation

Evaluation of individual components was performed during individual trials of the AP study and complete functionality available for use during the entire VNA study. The patient weight and heparin dilution were set with the GUI of the heparin control application. Manual verification of the measurement results and actions was performed with another GUI for evaluation and safety reasons. All other steps, like processing of the results and calculation of rate and bolus, were performed automatically. If a bolus was necessary, the caregiver was prompted a notification to verify sufficient amounts of fluid are present within the heparin syringe. Thereafter, the bolus was administered automatically by the system.

11.3 Results

Automated processing and calculation of the correct adaptions and heparin infusion rates was successfully performed for 125 measurements during the VNA study and adaption was performed according to the used therapeutic table at all times in the used experimental ICU setting. On average, each animal subject was kept at a mean ACT measurement time of 95.5 s. As an example, the therapeutic management for vna006 is shown in Figure 11.5. With this successful automation the desired workload and error reduction could been achieved.

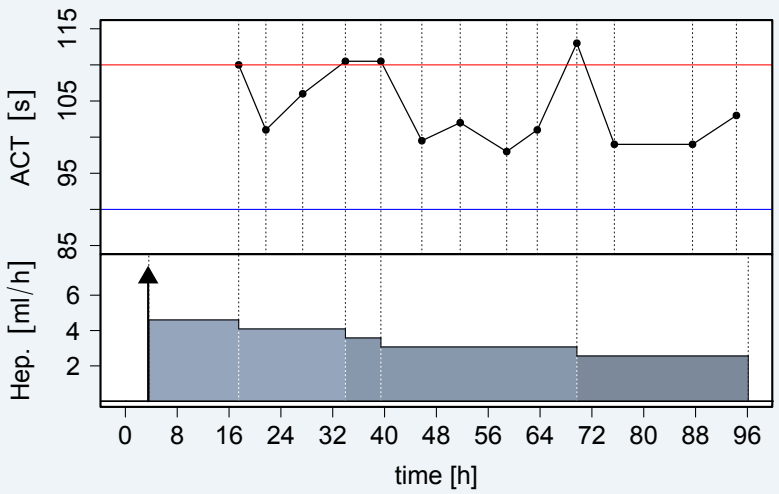


Figure 11.5: Example for the automated adaption of the heparin (Hep.) infusion rate (bottom) according to the measured mean ACTs (top) for vna006. The arrow indicates initial bolus administering.

11.4 Discussion

The proposed hard- and software provides a successful implementation for an automation of anticoagulation management using a Medtronic ACT Plus device not supplied with the required data export features. The branching out of the electrical signals from the 7-segment display of the Medtronic ACT Plus device provided a successful solution for automated gathering of the measured results as it is less prone to errors in comparison to image processing using machine learning where factors like physical obstructions during the measurement, sunlight, reflections, movement, and other things would have to be considered. Of course, automation could have been implemented without the performed extraction of the measurement results, however this would retain the crucial step of manual input of the results with the corresponding human error potential. The remaining steps of blood collection and sample processing at the ACT measurement device cannot be further automated with the given technology as handling of the blood samples still requires manual labor.

For evaluation purposes of the algorithm and automation pipeline, manual verification of the measurement results has been performed. It became clear that this step is not only useful for evaluation purposes but in general for such a critical parameter as anticoagulation, where handling errors may lead to severe consequences like thrombosis or internal bleeding. Thus, especially the administering of a bolus has been and should be verified. Whereas this still requires manual interaction, the work intensive task of collecting the measurement results, calculating the change of the infusion rate according to patient weight and drug dilution, and setting the actual continuous and bolus rates has been automated.
Given the achieved automation the risk for human error and the required workload can be

reduced significantly. This is especially beneficial in settings where understaffing is present or with junior staff who are still developing clinical assessment skills as such solutions support the decision-making process and provided additional safety [1].
Within the given study setting, the automation of anti-coagulation management achieved such a reduction of human error potential and workload and improved the management of hemostasis. Furthermore, it allowed the automated collection and storage of the ACT measurements, which previously could only be collected manually on a sheet of paper and were not integrated into the central database.

11.4.1 Outlook

For future improvements of automated anticoagulation management, other parameters and factors might be considered and included in the calculations. Adaption of the heparin rate shows that coagulation is highly dependent on the dilution and amount of body fluid. During the VNA studies with purposeful increase and decrease of the volume state, the clotting time is changing and the heparin infusion rate must be adjusted accordingly. For further, advanced versions of this automation, factors like infusion rates and vital parameters, exceeding the look-up table as hospital standard, may be considered. As ACT measurements are often influenced by various additional factors, such as hypothermia, haemodilution, thrombocytopaenia, and platelet inhibitors or antagonists [197], even further improvements may include measurements of the body fluid content as a derived variable of infusion rates, weight measurements, and body temperature. This could further be extended to a learning algorithm able to adapt to the patient's needs by observing the effect of rate adaptions and increasing or decreasing the changes by closed-loop feedback.

12 Temperature Management

12.1 Motivation

Besides the given example for automated processing of laboratory results and the ACT management, other parameters and procedures performed during intensive care can be automated as well. Another such example is management of patient temperature. As motivated in the background, temperature management is still not performed at all times within the operation room [23] and is often only a subsidiary task. Whereas invasive methods for blood warming can provide automated temperature management, more commonly and wide-spread applicable forced-air units with special blankets require manual observation and adaption of the device according to patient temperature. As most of the time, a TTM with a defined target temperature is desired [22], observations and adaptions have to be performed regularly. Forced-air warming devices only provide a limited number of settings, thus most often no optimal steady state can be achieved with those adaptions. Instead, continuous observation and corrections are needed to keep patient temperature within the desired range. Distractions or other more important therapeutic goals might restrict those observations or prolong adaption intervals when performed manually.
As task itself is simple – adapting airflow and temperature settings to keep a desired patient temperature – it's a good candidate for automation and workload reduction by reducing the need for manual observation and interaction. This may allow anesthetists or other care givers to focus on more important tasks and can reduce human errors causing hyper- or hypothermia due to delayed corrections. Within the context of conducted experimental studies, automated management of temperature was one important aspect to achieve more stable study conditions and more comparable results by reducing potential effects of hyper- or hypothermia to the desired homeostasis management and maintenance.
For application of temperature management, a very commonly used product is the 3M Bair Hugger for forced-air warming unit with different temperature and speed settings. Using the developed framework, temperature measurements of the patient were already available and closed-loop automation and control messages could easily be implemented. However, because the 3M Bair Hugger provides no interface except for manual buttons, an electrical interface for remote control had to be established. The performed development process and the evaluation were successfully published [198] and are presented below.

12.2 Hardware and microcontroller

The used 3M Bair Hugger 755 warming unit [199] provides settings for 3 defined temperature levels ($32.0 \pm 1.5\,^{\circ}\mathrm{C}$, $38.0 \pm 1.5\,^{\circ}\mathrm{C}$, and $43.0 \pm 1.5\,^{\circ}\mathrm{C}$) and ambient temperature.

Besides temperature, the airflow speed can be adapted with two speed settings: *Fast* (4700 rpm) or *Slow* (4100 rpm), resulting in a maximal airflow of 23 L/s [199]. As the used 3M Bair Hugger 755 provides no external interface for those settings, development of a hardware interface had to be performed as a first step.

Dismantling and analysis of the device and its integrated circuit boards revealed the electrical traces and logic levels of the 3M Bair Hugger's electrical connections, providing the essential knowledge for development of an interface for reading the current settings and providing remote inputs.

For processing the current state, the information of the lit LEDs of the 3M Bair Hugger device was sensed. This was performed by tapping the 10 LED outputs and using TLC 374 comparator ICs for conversion of the LEDs analog voltage after the limiting resistor to digital levels that can be processed. The information obtained from the LED signals includes fault state, temperature in range, and settings for current speed: standby, slow, or fast. The current temperature is shown with LEDs for ambient, 32 °C, 38 °C, and 43 °C settings [199]. For interaction with the 3M Bair Hugger's switches and simulation of manual key strokes, HFC 4066 ICs were used as digital switches and connected in parallel to the manual switches on the device's circuit board. In total, 8 output pins were used to control the 3M Bair Hugger switches. Preliminary tests indicated that a press time of 150 ms is a reliable time for detection by the manufacturer's logic board.

Given these hardware interfaces to the current device state by processing the LEDs and the ability to control the 3M switches, the next step consisted of connecting the individual components to a microcontroller for processing and communication.

For this task an Atmel AVR Mega8A [200] (Microchip Technology Inc.) microcontroller was used, as it provided the required number of input and output pins for reading the device states, controlling the switches, and a serial interface for information exchange. Software development was performed C using Atmel Studio 6 with GCC. By using timer interrupts, the current state of the 3M Bair Hugger's LEDs, connected to input pins via the TLC 374 comparators, is processed once a second and sent to the serial interface. Furthermore, received serial control commands are applied by simulating manual key presses using the HFC 4066 ICs connected in parallel to the actual 3M Bair Hugger's switches. The integration of the developed add-on board into the 3M Bair Hugger unit and the schematic circuit are shown in Figure 12.1.

12.2.1 Communication protocol

Communication is performed using a serial RS232 interface with a speed of 9600 Baud. Therefore, a matched Quartz crystal with 15.7456 MHz as the operation frequency of the microcontroller is used. The device state and settings are transmitted once a second in alphanumerical ASCII format and framed with a start and end character: `0x02` (STX) and `0x03` (ETX), respectively. The LED states are sent as numerical string of 0s and 1s. For remote control of the switches, a simple ASCII message of the format `S:X`, where `X` is the number of the Switch (1-8) that should be pressed for the pre-set time, can be transmitted.

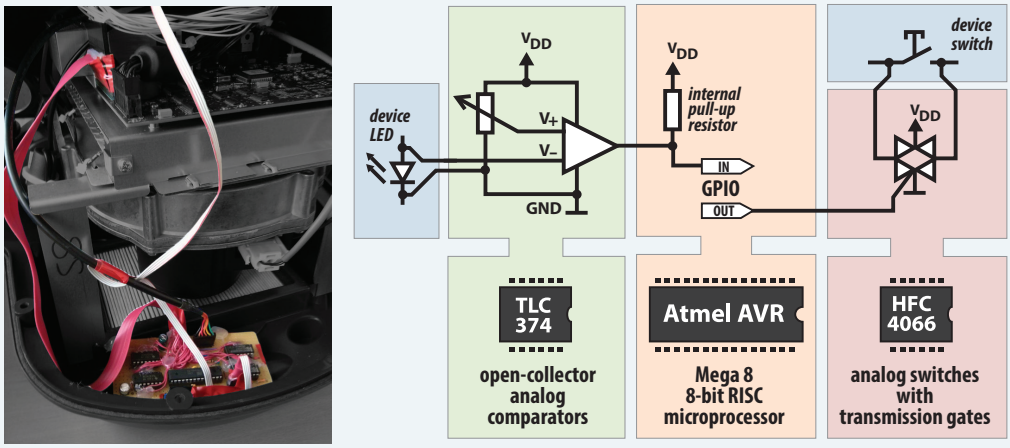


Figure 12.1: Opened 3M Bair Hugger 755 device with add-on board. (left, colors of original 3M Bair Hugger desaturated) [198] and illustration of the hardware interface (right)

12.3 Control algorithm

The proposed algorithm [198] for temperature control uses three parameters: an evaluation interval for the adaption c_t, a target temperature T_{tar}, and a scaling factor c_s for the power of adaption. Aggregation of individual measurements is performed during the defined interval c_t and upon evaluation the arithmetic mean of the collected measurements is used. This is performed using the current point in time t, the fixed interval c_t in seconds, target temperature T_{tar}, and the current body temperature T_t at time t:

$$\overline{T_t} = \frac{1}{c_t} \cdot \sum_{i=t-c_t}^{t} T_i$$

Given this averaged observation, in a next step the difference between the current observed measurement $\overline{T_t}$ and the target temperature is calculated:

$$\Delta_t = \overline{T_t} - T_{tar}$$

Based on this absolute change, a state $s_t \in [-1,1]$ at time t is calculated using the desired target Δ_t and the scaling factor c_s. This state variable s_t represents the action to be applied to the 3M Bair Hugger and is limited within the interval $[-1,1]$. Within this notation $s = -1$ stands for maximal cooling and $s = +1$ means maximal heating. At the initial step $t = 0$ the state variable is set to zero ($s_0 = 0$).

$$s_t = \begin{cases} 1 & \text{if} \quad \Delta_t \cdot c_s \geq 1 \\ \Delta_t \cdot c_s & \text{if} \quad -1 < s_t < 1 \\ -1 & \text{if} \quad \Delta_t \cdot c_s \leq -1 \end{cases}$$

Given this abstract representation s_t of the action to be performed, the final step consists

of a mapping to the device settings. The possible device settings are a tuple consisting of speed and heat setting $x_t = (speed, heat)$ with $speed \in \{standby, slow, fast\}$ and $heat \in \{21\,°C, 32\,°C, 28\,°C, 43\,°C\}$. The setting for ambient temperature is denoted as 21 °C, as this was the room temperature within the used experimental setting during the VNA studies. Beside those 3 speed settings, the 3M Bair Huggger has an additional *standby* setting. This is not used for the control, as this would deflate the forced-air blanket, which was considered impractical as it would influence the continuous monitoring of patient weight.

$$x_t = \begin{cases} (21\,°C, fast) & \text{if} \quad & & s_t = & -1 \\ (21\,°C, slow) & \text{if} \quad & -1 < & s_t \leq & -0.5 \\ (32\,°C, slow) & \text{if} \quad & 0.5 < & s_t \leq & 0 \\ (38\,°C, slow) & \text{if} \quad & 0 < & s_t \leq & 0.5 \\ (43\,°C, slow) & \text{if} \quad & 0.5 < & s_t \leq & 1 \\ (43\,°C, fast) & \text{if} \quad & & s_t = & 1 \end{cases}$$

As this tuple x_t is a direct representation of the 3M Bair Hugger's state, application can be performed with the developed interface board.

12.4 Evaluation

The evaluation of the proposed closed-loop algorithm and system for temperature management was performed in the experimental setting for two animal subjects of the VNA study. The 3M Bair Hugger 755 with forced-air blankets was connected via the developed hardware interface to the developed and described medical communication framework.
For integration into the medical software framework, the serial COM-plugin was used and an APP-plugin was developed. This application sends the current device state to the framework and applies control commands. A second application was used to implement the proposed control algorithm. It received the patient temperature and sends control messages. Patient blood temperature readings were obtained with a temperature sensor within an arterial PiCCO catheter. This sensor was connected to Philips IntelliVue MP 50 patient monitor. Using the developed patient monitor connection, vital data, including temperature readings, was processed by the framework and provided to the proposed temperature control algorithm. Assessment of the performance was done by comparison of two trials with no control (N), two manually controlled trials (M) and two automated trials (A). The uncontrolled reference (N) was obtained during the vna001 and vna002 trials. Manual control (M) was performed for vna003 and vna004. The automated algorithmic control (A) was performed for vna007 and vna008. The desired target temperature was set to 38 °C for both manual controlled (M) and automated (A) trials as this is a normothermic porcine temperature. For the uncontrolled trials (N), no active temperature management was performed. The used measurement setup and interaction of the devices is shown in Figure 12.2. Evaluation was performed in terms of how well the temperature could be kept at 38 ± 1 °C for the entire duration of observation.

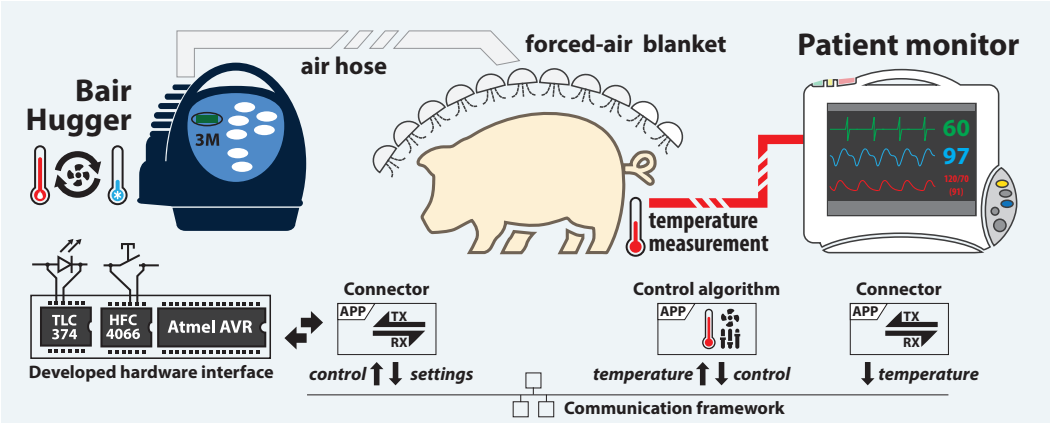


Figure 12.2: Experimental closed-loop TTM setup with 3M Bair Hugger and forced-air blankets for temperature management based on temperature observations obtained from a Philips IntelliVue MP50 patient monitor.

12.5 Results

Using the developed circuit, a 3M Bair Hugger 755 device was successfully extended by monitoring capabilities and remote control. The developed hardware interface was successfully integrated into the developed medical software framework and tested in the performed study. The results for the uncontrolled (N) and manual (M) reference animal subjects as well as the results for two automatically controlled animal subjects (A) of the VNA study are shown in Table 12.1. Whereas in the uncontrolled animal subjects, temperature raised and was only within the desired range of $38 \pm 1\,^{\circ}\mathrm{C}$ for about 26% of the time in N1 (vna001) and 71% in N2 (vna002). For the manually controlled animal subjects M (vna003, vna004), an average performance of 96.4% was achieved. Using the developed control algorithm for animal subjects A (vna007, vna008), the temperature could be kept within the desired range at all times.

Table 12.1: Performance for the individual animal subjects regarding the time exceeding the target temperature $\pm 1\,^{\circ}\mathrm{C}$ in seconds and the overall performance within the target range [198]. N = uncontrolled, M = manual, A=automated.

trial:		**N1**	**N2**	**M1**	**M2**	**A1**	**A2**
outside	$[s]$	243583	55522	579	23538	0	0
total	$[s]$	330060	192168	333490	333020	330960	329270
within	$[\%]$	26.20	71.11	99.83	92.93	100.00	100.00

An overview of the temperatures during the entire observation time for all animal subjects is shown in Figure 12.3. For the manually controlled animal subjects (M1, M2), infrequent temperature adaptions can be observed. For the automatically controlled animal subject (A1, A2), the tight temperature control with its frequent adaption steps can clearly be observed.

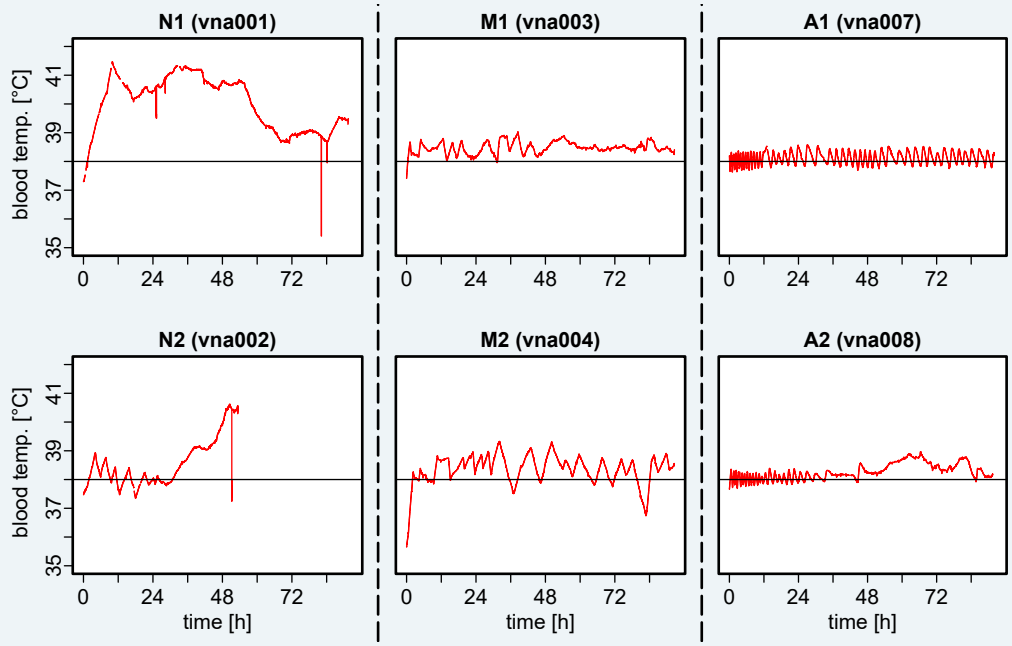


Figure 12.3: Temperature measurements for the two animal subjects of each group: uncontrolled (N), manually controlled (M), closed-loop controlled (A) in columns from left to right [198].

12.6 Discussion

Using a developed hardware interface, remote control and monitoring capabilities were successfully added to a 3M Bair Hugger 755 forced-air patient warming device. Integration to the medical software framework and a developed closed-loop algorithm for temperature control allowed a tight TTM. Using the automated system, temperature could be kept within the desired target temperature range of 38 ± 1°C at all times. Whereas manual control could achieve similar good results in the performed study, continuous human observation leading to a high workload was required.

For the two animal subjects without temperature management, large variations and durations outside the desired range could be observed. As trial N2 was cut short due to medical complications, further potential temperature increase could not be observed. Yet, it is expected to have been risen in a similar fashion as observed for animal subject N1. Thus, resulting in an equally decreased accuracy. As all animal subjects were observed in the same study conditions and the hyperthermia was observed in both uncontrolled as well as other previously studies animal subjects, this temperature increase was expected for the controlled

subjects as well. Thus, by keeping the temperature at the desired target level, successful application of temperature management for the automated and manual observations could be shown. By maintaining normothermia, the effects of body temperature changes in the managed animal subjects could be limited, thus minimizing potential temperature effects on the biochemical processes and improving the desired management of homeostasis.
The used algorithm provided a dynamic adaption to the animal subject's requirements as it does not rely on fixed temperature or speed settings, but performed an annealing to the desired target. Fixed settings or look-up table approaches would not suffice those individual needs as warm air, sufficient in one case, might be not enough to keep temperature level stable in another and lead to chilling. Hence, the only predefined setting is the desired target temperature level as suggested for TTM [22].

12.6.1 Outlook

Using automated temperature control, human workload and error potential can be reduced and temperature management may even be performed in settings where no human supervision is present at all times. Additionally, an automated system with closed-loop feedback from a patient monitor can provide more dynamic and faster adaptions, as it is only monitoring a single patient and no frequent human observation and interaction is needed.

As temperature management is an important aspect for keeping the body in a steady and stable state and improve patient outcome [30–35] but still not applied in all ORs [23], such an automated system may lead to a more widespread temperature management. For clinical application of course, further improvements and security measures are needed. Especially the serial communication and interface to the device was kept simple for this proof-of-concept approach. For clinical application, an enhanced protocol with redundancy, validation and check-sums should be used. However, such features could easily be integrated by the device manufactures and not be provided as aftermarket additions.

An improved algorithm may use further patient monitoring information to adapt to the patient need. For example, multiple sensors of different types, like placed on the skin or integrated in catheters [24, 25] may be used and combined to account for the different variable factors of heat loss [25] and include factors like body mass that may influence the response to heating and chilling.

13 Management based on fuzzy logic

13.1 Motivation

Besides anticoagulation management and temperature control to improve hemo- and homeostasis management and automate manual tasks with high workload and human error potential, the ability to achieve closed-loop control of individual physiological parameters for further improved management was an important aspect for the design and implementation of a fuzzy control system.
As shown throughout this thesis, homeostasis management is a crucial aspect of critical care, including various vital parameters, other factors, and environmental conditions to be kept within desired ranges by a manifold of used therapeutic protocols and drugs. Whereas many model-based and specialized approaches for the management of individual parameters exist, combining different and independently developed control systems, often with fixed settings and parameters, is difficult. To evaluate the interactions and to achieve the desired management, a more universal solution for usage in the experimental setting is therefore needed. A further complication to be considered is the high variability and dynamics of critical care settings as the ensemble of used medical devices and administered drugs is highly depending on the individual needs of each patient. Thus, a fixed system with limited and static closed loops would not suffice to provide enough flexibility outside of an isolated setting. For studying a broad variety of clinical research questions and conditions, mainly two important aspects needed to be considered:
First, a system should be flexible enough to accept arbitrary numerical inputs of vital parameters and provide outputs for the adaption of infusion rates or ventilator setting according to targets set by physicians. And second, manual therapeutic interactions and interferences have to be considered and must be tolerated by the system.
Regarding the first problem, the system needed to be designed in such a way, that the control function can be described by relevant measures including target level, critical physiological boundaries, and temporal aspects familiar to caregivers. As therapeutic guidelines often are designed as linguistic *if* ... *then* rules of fuzzy nature [7], a general goal-directed system using the concept of fuzzy logic, as introduced in the background of this thesis, allowed for adaption by physicians and caregivers prior to and during therapy.
To solve the second problem, closed-loop control had to be performed in a way such that human interaction was not prohibited. Given a calculated action to be applied, human interference with the system must be considered at any time, especially in critical conditions where infusion rates or ventilator settings may be adapted manually. Thus, no internal state or absolute values could be used, as they may become void at any time. Instead, an adaptive system based on relative output factors was considered and implemented, allowing an annealing to the desired parameter instead of setting absolute targets.

13.2 Fuzzy logic controller

The developed fuzzy controller used the general principle described in the background of this thesis. Numerical input values are fuzzified, processed by a set of rules, and defuzzified to provide a numerical output. However, before any fuzzy control can be performed, the fuzzy system itself needs to be configured. These individual steps should now be highlighted in detail for the specific implementation within the clinical context. As rapid changes in patient conditions with immediate need for action may occur, the dynamic response of the system and the ability for manual interferences and adaptions were essential considerations. A controller based on absolute values and targets would not allow for manual interactions and prohibit the physician from performing changes and interactions in critical, life-threatening situations.

13.2.1 Overview

The used controller concept was implemented for single input and output relations but can be expanded to multiple inputs. The controlled variable can be any vital value or processed parameter. This controller input is processed by a fuzzy function and the first derivative is calculated to obtain a second fuzzy function, representing the change of the input parameter between the current and last obtained measurement value. As for a first measurement no previous measurement is available, a slope of zero is defined for this step. Depending on the frequency of data acquisition, adaptions for every single new measurement may be too frequent, e.g., for data collected from the ventilator once a second. Thus, values can be aggregated within the controller and an averaged value is considered for a single control step when a user-definable period of time has passed. On contrary, for infrequent or event-based parameters like those obtained from BGA, the system may be set up to perform adaption for any new measurement. The output of the controller is modeled as a relative factor to adapt the current setting or state of an actuator like infusion pumps or ventilator. In clinical applications, manual therapeutic interactions or other automated measurements may be performed and interference with the control algorithm. To avoid such disruptions, e.g., during therapeutic maneuvers or respiratory hold, the fuzzy controller can be locked by a special message. Within this locked state, no measurements are calculated and no adaptions are performed. As the proposed fuzzy controller is a general-purpose application for arbitrary vital parameters, the relation between the measured and controlled parameter may be inverse. Given a high glucose level, the infusion rate should be reduced, whereas given a high $etCO_2$ reading, RR should increase uni-directionally. Thus, this relation is an additional parameter for a specific fuzzy controller. Further user-definable parameters are the type and parameter of an input message and the output to be controlled. An in-dept description of those individual parameters is given in subsequent sections. In short, the input and output may be any message with the key-value based MDT format or direct control command for ventilation device or pumps. Furthermore, a target value c_t with critical upper and lower boundaries (c_{cl},c_{ch}) and the operation mode (interval or event-based) need to be set. For considering the input parameter's slope, a weighting factor w_Δ is used. Additionally, time frames, defining lower temporal boundaries for a parameter to reach one of the critical

boundaries (t_{lc},t_{hc}) from the desired target level, need to be set. For adaption of the output strength, a scaling factor a is used. This general pipeline with the used parameters and steps is illustrated in Figure 13.1.

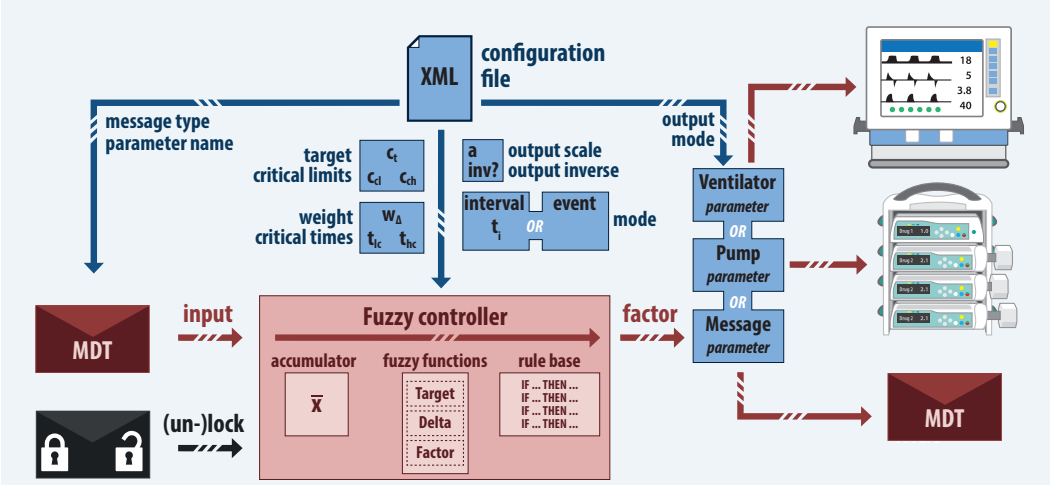


Figure 13.1: Illustration of the implemented general purpose fuzzy logic controller utilizing relative output factors. Input is provided by message objects and processed according to user-defined target levels and boundaries. The relative output can directly be sent to infusion pumps and ventilator or provided as a message object. To avoid undesired adaptions, the controller can be locked and unlocked with a special message at any time.

13.2.2 Modeling of the controller

Modeling of the fuzzy logic controller of Mandani type is performed using two input variables and one output variable. Each variable consists of five member-functions representing *critical low*, *low*, *normal*, *high*, and *critical high* states. To provide a smooth control behavior with a relatively flat slope around the target value, Gaussian and sigmoid curves are used for modeling of the fuzzy member functions. For the set operations of the different processing steps within the fuzzy controller the following operators are used: set union – *max*; intersection – *min*; implication – *min*; accumulation – *prod*; defuzzification – *centroid*. As already motivated, the input for this controller and the fuzzy sets should be limited to parameters that can easily be defined by physicians. Therefore, only a desired target (c_t), critical upper and lower bounds (c_{cl}, c_{ch}), time constants (t_{lc},t_{hc}), and a weight factor w_Δ are required as user-input. All other parameters like the individual membership functions of the fuzzy functions are automatically calculated based on those settings.

13.2.2.1 Symmetry and scaling considerations

As the developed fuzzy controller is using an output factor instead of a defined range, this factor needs to be calculated and processed by the fuzzy logic controller. For factorial output, 1 refers to no change of the target parameter, whereas 2 results in a doubling, and 0.5 in a bisection. In more general terms, increment of factor x is equal to the same

decrement with a factor $\frac{1}{x}$. This reciprocal relationship must be mapped to the fuzzy set to provide a symmetric response. Simply using the fractions results in a skewed distribution of the member functions and numerical inaccuracies due to rounding. Especially for small fractions, this effect needs to be considered for technical implementation. To overcome this problem, the numerical ranges for increasing and decreasing factors should be equal: An increment of x should correspond to its equal decrement with $-x$, therefore resulting in equal numerical ranges at all times for possible parameter inputs. This is achieved by mapping the factor into exponential space using 10^x $(x : abs(x) > 1)$ as the representation within the fuzzy set. Given an exemplary factor of $x = 2$ using 10^x, $10^2 = 100$ equals 100 times more, whereas $10^{-2} = 0.01$ equals 100 times less. Figure 13.2 illustrates the behaviors and differences in the resulting numerical ranges for normal and exponential scaling of the output factor calculation. To allow for scaling of the individual fuzzy sets in the implementation, an additional scaling factor a is introduced. The fuzzy set for the output factor therefore is represented as 10^{ax} within the proposed controller.

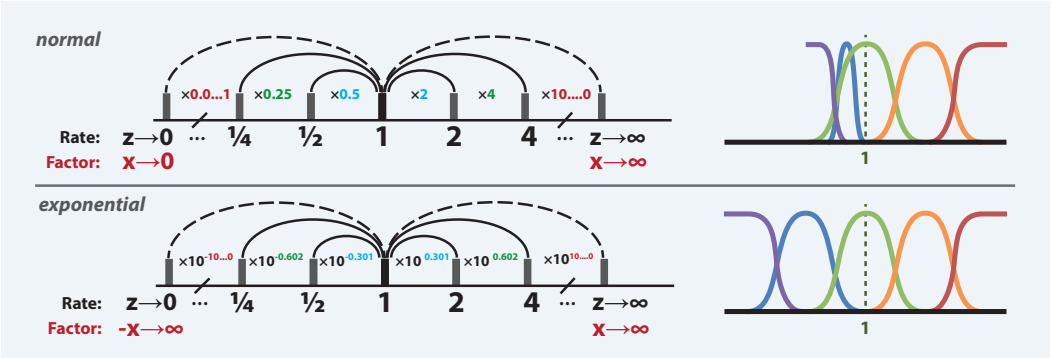


Figure 13.2: Illustration of the advantage of the exponential factor used within the fuzzy logic calculation. The numeric ranges for large and small changes are equal in the exponential modeling, resulting in a unscrewed fuzzy set.

13.2.2.2 Time to criticality

As already highlighted, the developed fuzzy controller uses a weighted combination of absolute value and slope for the calculation of the output factor. Yet, establishing values for slopes that reflect clinical knowledge and guidelines is difficult and the question to be asked for configuration of the fuzzy controller is: How can physicians assess critical slopes if the numerical value for such a parameter change is neither common knowledge nor a practical clinical measure? However, given clinical practice, time spans are more natural concept and often well known. This allows for a more practical approach by defining time periods in which the parameter should not become critical, like a shift of 8 hours, after which the next routine measurement may be performed. Thus, instead of defining a critical slope, a measure for the time it takes for an observed value to reach one of the critical boundaries from the desired target level is introduced and denoted as time to criticality (TTC). It combines the physiological limits set by the physician with the temporal domain and, given asymmetric risks, different lower and upper TTC may be required. For usage within the fuzzy controller, the remaining step is to convert those times to critical slopes s regarding

to the input parameter. Calculation of the critical slopes s_{lhc} for the lower and s_{thc} for upper bounds are based on the target value c_t, the critical boundaries c_{lc}, c_{hc} and the TTC for both boundaries, t_{lc} and t_{hc}.

$$s_{tlc} = \frac{c_{lc} - c_t}{t_{lc}} \qquad s_{thc} = \frac{c_{hc} - c_t}{t_{hc}}$$

A visual illustration of this process for slope calculations using the use-defined critical times is shown in Figure 13.3.

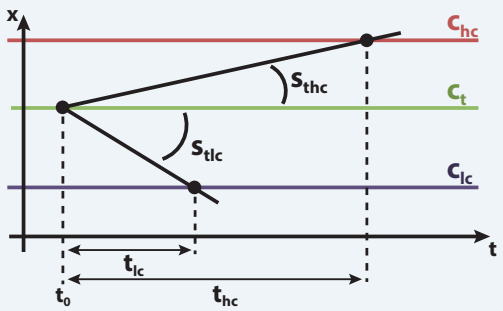


Figure 13.3: Illustration of the time to criticality (TTC) calculation as the slope for reaching the critical boundaries in a defined duration of time.

13.2.2.3 Input and output variables

The fuzzy sets of the developed proportional fuzzy logic controller consist of 5 membership functions for each variable. The arrangement of the membership functions for the individual sets was designed to provide a scalable distribution for changing parameter ranges as the fuzzy set will be applied to various parameter ranges, often unknown in advance and therefore cannot be scaled or normalized.

For the implemented fuzzy controller, only two types of membership functions m are used: Sigmoid and Gaussian, denoted as $sigm(c,s)$, with center c and slope s, and $gaus(c,sd)$, with center c and standard deviation sd, respectively.

The first input variable F_{TARGET} consists of 5 membership functions and is used for processing the measurement input and requires the user-defined inputs for target c_t and the two critical boundaries c_{lc} and c_{hc}. The calculation is illustrated in Figure 13.4 and defined as

follows:

$$
\begin{aligned}
f_{TARGET} &= \{\mu_{LC}, \mu_L, \mu_N, \mu_H, \mu_{HC}\} \\
with: \quad \mu_{LC} &= sigm\left(c_t - (c_t - c_{lc}) \cdot 0.8, -100 \cdot (c_t - c_{lc})^{-1.5}\right) \\
\mu_L &= gaus\left(c_t - (c_t - c_{lc}) \cdot 0.5, (c_t - c_{lc}) \cdot 0.15\right) \\
\mu_N &= gaus\left(c_t, \frac{(c_t - c_{lc}) + (c_{hc} - c_t)}{2} \cdot 0.2\right) \\
\mu_H &= gaus\left(c_t + (c_{hc} - c_t) \cdot 0.5, (c_{hc} - c_t) \cdot 0.15\right) \\
\mu_{HC} &= sigm\left(c_t + (c_{hc} - c_t) \cdot 0.8, 100 \cdot (c_t - c_{lc})^{-1.5}\right)
\end{aligned}
$$

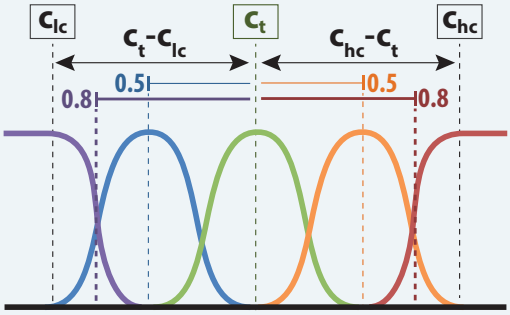


Figure 13.4: Illustration of the calculation for the generation of the fuzzy sets with their membership functions from the small set of input variables.

The second input variable F_{DELTA} consists of 5 membership functions and is used for processing the slope of the measured input. For the first measurement, a slope of zero is used. For definition of the fuzzy set, the input parameters for target c_t and the two critical boundaries c_{lc} and c_{hc} as well as the two TTCs t_{lc} and t_{lc} are used and the input fuzzy function is defined as follows:

$$
\begin{aligned}
f_{DELTA} &= \{\mu_{DLC}, \mu_{DL}, \mu_{DN}, \mu_{DH}, \mu_{DHC}\} \\
with: \quad \mu_{DLC} &= sigm\left(s_{tlc}, s_{tlc}^{-1.5}\right) \\
\mu_{DL} &= gaus\left(s_{tlc} \cdot 0.5, s_{tlc} \cdot 0.2\right) \\
\mu_{DN} &= gaus\left(0, \frac{s_{tlc} + s_{thc}}{2} \cdot 0.2\right) \\
\mu_{DH} &= gaus\left(s_{thc} \cdot 0.5, s_{thc} \cdot 0.2\right) \\
\mu_{DHC} &= sigm\left(s_{thc}, s_{thc}^{-1.5}\right)
\end{aligned}
$$

The output function f_{FACTOR} for the output factor consists of 5 membership functions with

a constant definition:

$$
\begin{aligned}
f_{FACTOR} &= \{\mu_{DECH}, \mu_{DEC}, \mu_{KEEP}, \mu_{INC}, \mu_{INCH}\} \\
with: \quad \mu_{DECH} &= sigm(-0.7, -15) \\
\mu_{DEC} &= gaus(-0.4, 0.2) \\
\mu_{KEEP} &= gaus(0.0, 0.2) \\
\mu_{INC} &= gaus(0.4, 0.2) \\
\mu_{INCH} &= sigm(0.7, 15)
\end{aligned}
$$

13.2.2.4 Rule base for implication

The two input variables for absolute value (*Target*) and slope (*Delta*) are evaluated to obtain the output factor (*Factor*) of the proportional fuzzy logic controller using the fuzzy logic concept of simple if-then rules. Rules for absolute value and slope are defined independently with the user-definable weight factor w_Δ for balancing the influence of absolute value (*Target*) and relative change (*Delta*). As all rules are evaluated by the fuzzy controller and already combined in the aggregation step, no combinations of both input variables were explicitly implemented in an extended set of rules. Dependent on the variable to be controlled and the action to be performed, one of two rule sets can be selected by the user. The first one is a unidirectional change of the input variable in relation to the output variable. An increment of the measured value thus results in an increment of the output factor. On contrary, the second set of rules implements an inverse relation. Increase of the input results in a decrease of the output factor. This second relation is of clinical relevance for many parameters, where the infusion rate should be reduced if the corresponding measurement is increasing. This inverted rule base for the developed fuzzy controller is shown in Figure 13.5. For the non-inverted rule base, the roles of INCH/DECH and INC/DEC are exchanged, respectively.

If *Target* **is** *LOWCRIT*	**then** *Factor* **is** *INCH*	**with**	$1 - w_\Delta$
If *Target* **is** *LOW*	**then** *Factor* **is** *INC*	**with**	$1 - w_\Delta$
If *Target* **is** *NORM*	**then** *Factor* **is** *KEEP*	**with**	$1 - w_\Delta$
If *Target* **is** *HIGH*	**then** *Factor* **is** *DEC*	**with**	$1 - w_\Delta$
If *Target* **is** *HIGHCRIT*	**then** *Factor* **is** *DECH*	**with**	$1 - w_\Delta$
If *Delta* **is** *DELTALOWCRIT*	**then** *Factor* **is** *INCH*	**with**	w_Δ
If *Delta* **is** *DELTALOW*	**then** *Factor* **is** *INC*	**with**	w_Δ
If *Delta* **is** *DELTANORM*	**then** *Factor* **is** *KEEP*	**with**	w_Δ
If *Delta* **is** *DELTAHIGH*	**then** *Factor* **is** *DEC*	**with**	w_Δ
If *Delta* **is** *DELTAHIGHCRIT*	**then** *Factor* **is** *DECH*	**with**	w_Δ

Figure 13.5: Used set of rules of the fuzzy logic controller with inverted output relation.

13.2.3 Implementation

The fuzzy controller was implemented in C++ as an APP-plugin within the TICoMS framework using the Fuzzylite library [201]. Each used controller is defined within the application's XML-based configuration file. The input parameter to be used are defined by a unique message and parameter name. The numerical output can either be set to a specific infusion pump or ventilation device parameter, or also be a defined parameter of a message in general MDT format. Configuration of the fuzzy controller is performed within the XML file by defining a desired target level $tar \in \mathbb{R}$ and lower $c_{low} \in \mathbb{R}$ and upper $c_{high} \in \mathbb{R}$ critical bounds. If calculation should not be performed for every measurement but averaged, the interval $t_{eval} \in \mathbb{N}$ must be set in seconds. For definition of the slope, the weight of the slope $w_{\Delta} \in [0,1]$ relative to the absolute value must be set. The critical boundaries for this slope are set as times to criticality t_{lc},t_{hc} in seconds for lower and upper bounds, respectively. For easy adaption of the regulatory power, a scaling factor $f_a \in (0,1]$ for the calculated output needs to be set. Finally, the general relation between the input and output parameter is defined with a flag to indicate if the relation between two factors follows an inverse rule. If this flag is set, an increase of the input variable results in a reduction of the output factor.

13.2.4 Evaluation

For evaluation of the proposed fuzzy controller, parameters using event- and interval-based controls were implemented and tested during the two conducted studies. For the former operation mode, respiratory management was performed. The latter, event-based mode was applied to BGA measurements for homeostasis control using the infusion pumps. Parameter selection was based on the requirements during the studies and implemented for those examples.

To prohibit interference of various maneuvers and manipulations during the conducted studies, the fuzzy sets were automatically or manually locked during those interactions to prohibit calculations based on falsified or artificial readings.

For the animal subjects of the AP study, a first version of the controller, not yet implementing the 10^{ax} concept and TTC was used. For the animal subjects of the VNA study, those improvements were used for glucose management. The following sections present those individual applications and their respective results, concluding with a general discussion of the closed-loop management based on the proposed fuzzy logic controller.

13.3 Respiratory management

As highlighted in the medical background, management of homeostasis is an uttermost important aspect in critical care. Within this process, patient's respiration has a significant influence to achieving this goal, thus making respiratory management an essential factor for prevention of conditions like respiratory acidosis or alkalosis. In the context of medical studies, respiratory management additionally plays an important role to allow for measurement of other vital parameters in well-defined conditions.
The importance of ventilation management can be seen in Figure 13.6 for animal subject ap004. Given a relatively constant RR due to mechanical ventilation under GA and changing metabolic conditions, the body is not able to adapt the RR on its own. Thus, the CO_2 level in the exhaled air is highly variable and rising throughout the observation time. As the expiratory CO_2 is related to the intravascular levels and the acid-base balance, it has a significant influence on homeostasis. Therefore, by keeping the $etCO_2$ level stable, a more natural response in breathing and homeostasis management can be performed. Using this relationship between the RR and the measured CO_2 levels, automated management with the developed fuzzy logic was realized.

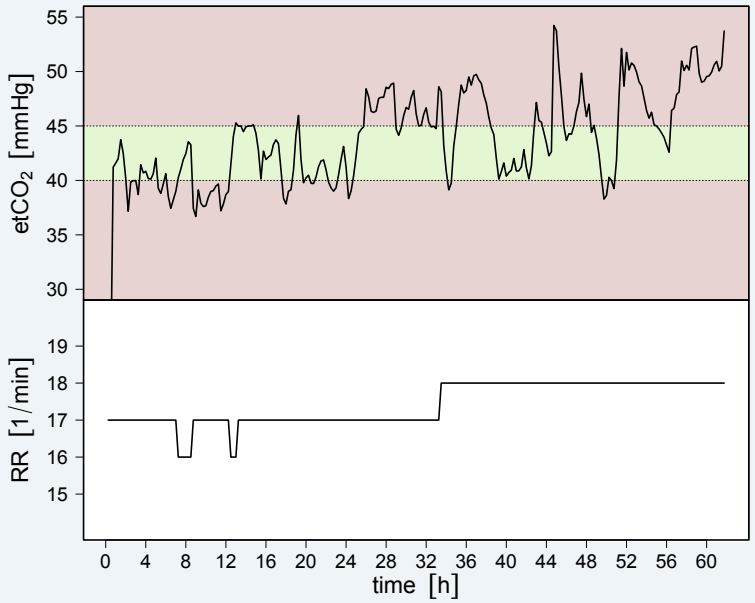


Figure 13.6: Example for uncontrolled $etCO_2$, often reaching levels outside the desired physiological range of 40–45 mmHg in animal subject ap004 with a relatively constant RR.

13.3.1 Setup

The used setup for automated management of $etCO_2$ consisted of the used mechanical ventilation device (Dräger Evita XL) with remote control capability and a bedside computer system integrated into the TICoMS framework. The ventilator was connected to an embed-

ded PC via serial connection for reading the measurement results and settings about once a second, and sending control commands to the device using a framework APP-plugin. The obtained measurements were sent via TICoMS as messages to the fuzzy controller application and collected for a defined interval I. After this predefined period has passed, the mean was calculated and evaluation of the measured $etCO_2$ level was performed. The resulting adaption factor for the RR was sent as control commands to the ventilator application and automatically applied. An illustration of this measurement and control setup is shown in Figure 13.7. This adaption of $etCO_2$ levels was performed for both studies, starting with animal subject ap005. For the AP study, an interval $I = 240s$ was used. This interval was reduced to $I = 120s$ in the VNA study to provide a faster response time. The desired $etCO_2$ target level t was set to a physiological level of 40 mmHg.

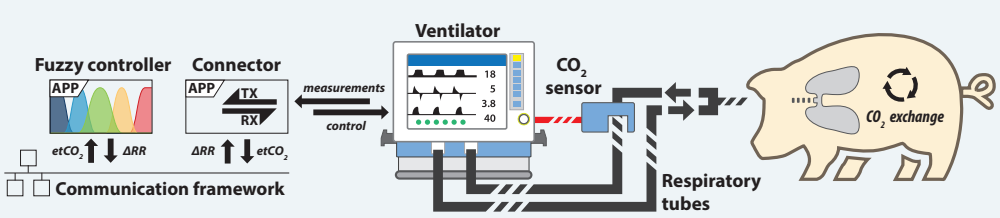


Figure 13.7: Schematic illustration of the fuzzy-controlled closed-loop setup for $etCO_2$ management by adaption of the respiratory rate.

13.3.2 Results

The automation setup was successfully implemented and tested during the performed studies. However, for three animal subjects (ap005, ap007, vna003) medical conditions, unrelated to the application of the closed-loop system, forced an adaption of the target $etCO_2$ level to 45 mmHg. Yet, management was still hardly possible. For animal subject ap010 unreliable measurements of the ventilator's $etCO_2$ sensor prohibited the closed-loop application. Thus, manual control and correction was performed by monitoring of oxygen saturation and arterial pCO_2 levels from BGA. This problem occurred repeatedly during the observation of another animal subject (vna002), whereupon the sensor was replaced. The results of the closed-loop control are visualized in Figure 13.8. For better visual presentation and reduction of measurement artifacts, the charts are plotted using mean averaged blocks of 120 seconds. Data falling within periods of manual intervention, e.g., respiratory maneuvers in the VNA study, is excluded and the last known value is imputed for visualization. For the boxplots, raw measurements are used. Whiskers are placed at the outermost values falling below or exceeding the interquartile range of Q1 to Q3 by 1.5 times, respectively. Numerical results for the individual animal subjects, like detailed statistics of the information within the figure's boxplots, targets, and deviations thereof, are provided in Table 13.8. Automated control could be performed for total duration of over 1066 hours and measurements within a range of ± 3 mmHg related to the set target could be observed in over 95% of the total observation and management time. Broken down into the two performed studies, management was successfully performed in 91% of the time for the AP study and the desired target range was maintained in 97% of the time within the VNA study.

Table 13.1: Configuration and results of respiratory management for the individual animal subjects with defined target $etCO_2$ level t, control interval I, and duration d.

	c_t [mmHg]	I [s]	d [h]	$t \pm 3$ [%]	rel. dev. [%]	IQR [mmHg]	MAD [mmHg]
ap005	45	240	37.3	0.85	1.72	1.78	1.23
ap006	40	240	32.4	0.65	3.30	4.82	2.01
ap007	45	240	56.0	0.90	0.83	0.73	0.53
ap008	40	240	72.0	0.98	-0.22	1.03	0.76
ap009	40	240	70.4	0.99	0.04	0.87	0.59
vna001	40	120	93.1	0.96	0.00	1.10	0.88
vna002	40	120	33.1	0.66	-21.61	12.68	1.48
vna003	45	120	94.1	0.96	-0.42	1.17	0.86
vna004	40	120	94.1	1.00	-0.51	0.90	0.80
vna005	40	120	93.7	0.95	-0.24	1.53	1.23
vna006	40	120	61.0	0.96	-0.21	1.22	1.01
vna007	40	120	93.4	1.00	-0.92	1.23	1.18
vna008	40	120	93.3	1.00	-0.74	1.27	1.06
vna009	40	120	93.3	1.00	-0.70	1.12	0.99
vna010	40	120	47.1	0.97	0.23	1.15	0.86

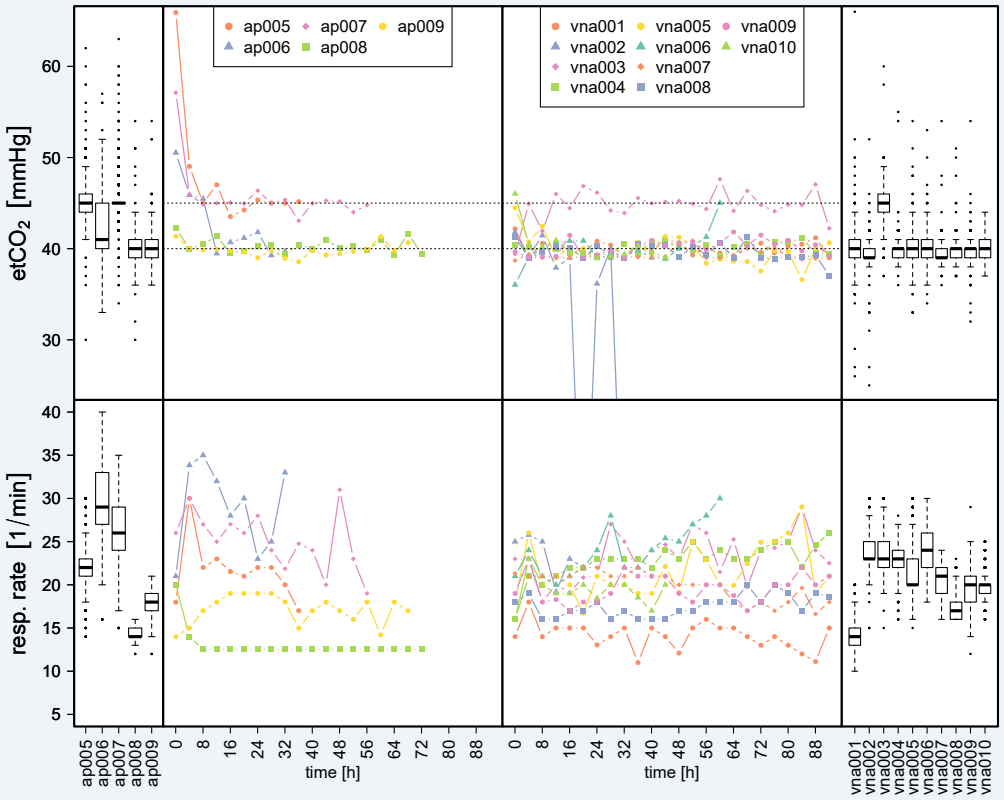


Figure 13.8: Observed $etCO_2$ levels (top) and RRs (bottom) for the individual animal subjects as time-series and boxplots for both used intervals (240s left, 120s right).

13.4 Glucose and electrolyte management

Besides respiratory management, maintaining glucose and electrolyte levels is another important aspect to preserve homeostasis and prevent conditions like hypoglycemia or metabolic conditions like alkalosis or acidosis. As treatment of the animal subjects is performed according to medical standards and required to obtain reliable results, management has initially been performed manually. To reduce workload and provide more stable conditions for collection of research data, management of different metabolic parameters was automated using BGA results to control infusion pumps with selected electrolyte and glucose solutions. Due to the prolonged observation of the animal subjects for several days, nutrition supplement by administering glucose and management of base excess were the most important parameters. Other parameters like calcium and potassium were evaluated only in a limited number of animal subjects if individual clinical indication was given. As the proposed fuzzy logic controller was used for all those automated therapies, first, the general setup and individual configurations for those parameters are presented.

13.4.1 Setup

For automated management of metabolic parameters, the BGA results were obtained from the Radiometer ABL 800 Flex analyzer using an APP-plugin within the developed framework. After validation by a caregiver to catch potential measurement errors, the collected results were automatically sent as message objects via TICoMS to the individual fuzzy controller application. After evaluation, the output factor was sent as a message to control application for the B.Braun Space infusion pumps. A schematic illustration of this control setup is given in Figure 13.9. The individual configuration of the evaluated fuzzy logic controllers is presented in Table 13.2. For glucose management, an improved version of the fuzzy controller with consideration of slope and criticality times was introduced for the VNA study, whereas in the AP study only the absolute measurement was used. All implemented controllers were event-based and triggered by new BGA measurements, without any aggregation and averaging. The controlled pumps were glucose (G20, 20%), NaBic (8.4%; 1000 mmol/ml), KCl (1 mmol/ml) and CaCl (5.5%; Ca^{2+} 0.5 mmol/ml, Cl^- 1 mmol/ml).

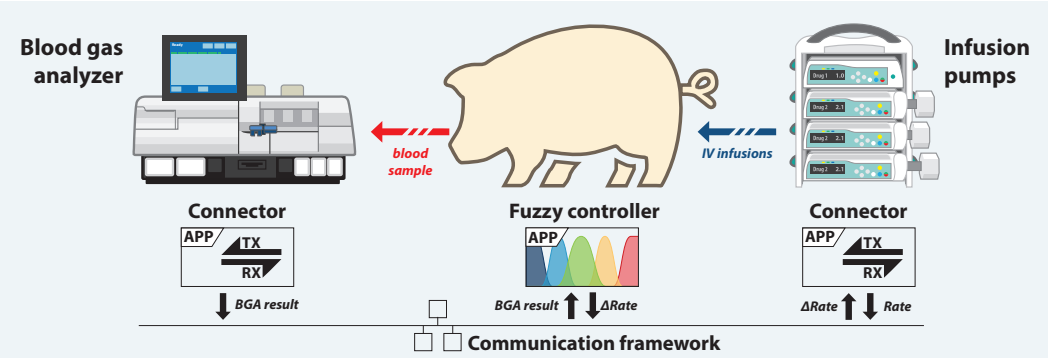


Figure 13.9: Schematic illustration of the fuzzy-controlled closed-loop setup for homeostasis management by adaption of infusion pumps based on BGA measurements.

Table 13.2: Configuration of the fuzzy logic controller for the evaluated parameters. Target parameters are arterial BGA measurements. Targets for glucose in mg/dL, potassium and base excess in mmol/L.

name	tar.	pump	c_t	c_{cl}	c_{ch}	w_δ	t_{lc}	t_hc	a	inverse?
Glu AP	Glu	G20	100	50	200	0	–	–	0.3	yes
Glu VNA	Glu	G20	100	50	200	0.3	12 h	1 h	0.3	yes
Base excess	BE	NaBic	0	-4	4	0	–	–	0.3	yes
Potassium	K^+	KCl	4	3	5	0	–	–	0.3	yes
Calcium	Ca^{2+}	CaCl	1	-0.8	1.2	0	–	–	0.3	yes

13.4.2 Results

The measurement for all metabolic parameters were based on arterial BGA measurements. In the AP study, BGAs were performed regularly in two-hour intervals whereas in the VNA study, measurements were performed event based for each change of 200 g BW. This resulted in a variable measurement frequency. Glucose management was automated for all animal subjects where supplementation was needed (ap005-009, vna001-010). Base excess and potassium were controlled, if clinical indication for supplementation was given. Only then, the corresponding electrolyte solutions were connected and the closed-loop controllers were started. Calcium was only managed during the AP study.

13.4.2.1 Glucose

The proposed fuzzy-controller was successfully implemented and tested in the performed studies with 15 animal subjects. For the AP study, a symmetric control a with target arterial glucose level $t = 100$ mg/dL was used and the slope was not considered. For the VNA study, asymmetric risk with lower and upper critical levels of 50 mg/dL and 200 mg/dL, respectively, were used with the same target level t. The slope was considered with asymmetric TTC for lower and upper bounds ($t_{lc} = 12$ h, $t_{lc} = 1$ h). Thus, the lower critical level of 50 mg/dL should not be reached within 12 hours, starting at the desired target level of 100 mg/dL.

An overview of the results is given in Table 13.3. Glucose levels could be kept within a physiological range of 86–150 mg/dL in 85 % (SEM 3.6 %) of the measurements with a total duration of 1142 hours. Regarding the individual settings, this goal could be kept in 83 % (SEM 7.8 %) and 86 % (SEM 4.0 %) of the time for AP and VNA study, respectively. Additional visualization of the measured levels and pump rates for all animal subjects is provided in Figure 13.11.

In two animal subjects (ap006, vna010) with short observations, management was only successful in 55 % of the time, whereas successfully performed in over 80 % for the remaining 13 animal subjects. Hypoglycemia was only present in 9 of 1141 hours for 5 animal subjects at the beginning of the observation. An example for the variable and individual need and successful glucose control by the enhanced controller is shown in Figure 13.10 for animal subject vna003, maintaining the physiological range in 98 % of the time.

Table 13.3: Results of glucose management for individual animal subjects with a target of 100 mg/dL

	glucose ranges [mg/dl]						
	≤70	[71,85]	(85,120]	(120,150]	>150	hours	within (85,150]
ap005	1	14	39	11	0	65	0.77
ap006	2	12	11	6	0	31	0.55
ap007	0	0	70	1	0	71	1.00
ap008	0	6	67	0	0	73	0.92
ap009	0	7	62	0	0	69	0.90
vna001	0	14	61	14	3	92	0.82
vna002	2	10	23	19	0	54	0.78
vna003	0	2	84	6	0	92	0.98
vna004	0	6	68	19	0	93	0.94
vna005	1	10	61	21	0	93	0.88
vna006	0	2	74	12	4	92	0.92
vna007	0	6	82	1	0	89	0.93
vna008	3	1	87	2	0	93	0.96
vna009	0	13	66	7	4	90	0.81
vna010	0	17	18	6	3	44	0.55
total	9	120	873	125	14	1141	**0.87**

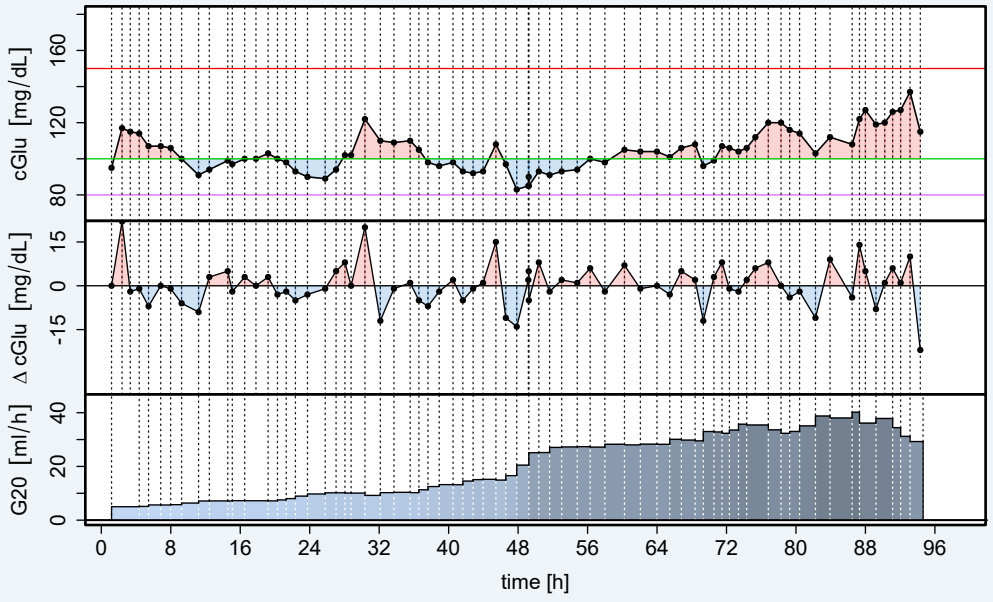


Figure 13.10: Example for successful glucose management for animal subject vna003. Shown are the measured glucose level (cGlu), the relative change (ΔcGlu) and the G20 infusion rate. The reaction of the controller to slope and absolute changes can clearly be observed.

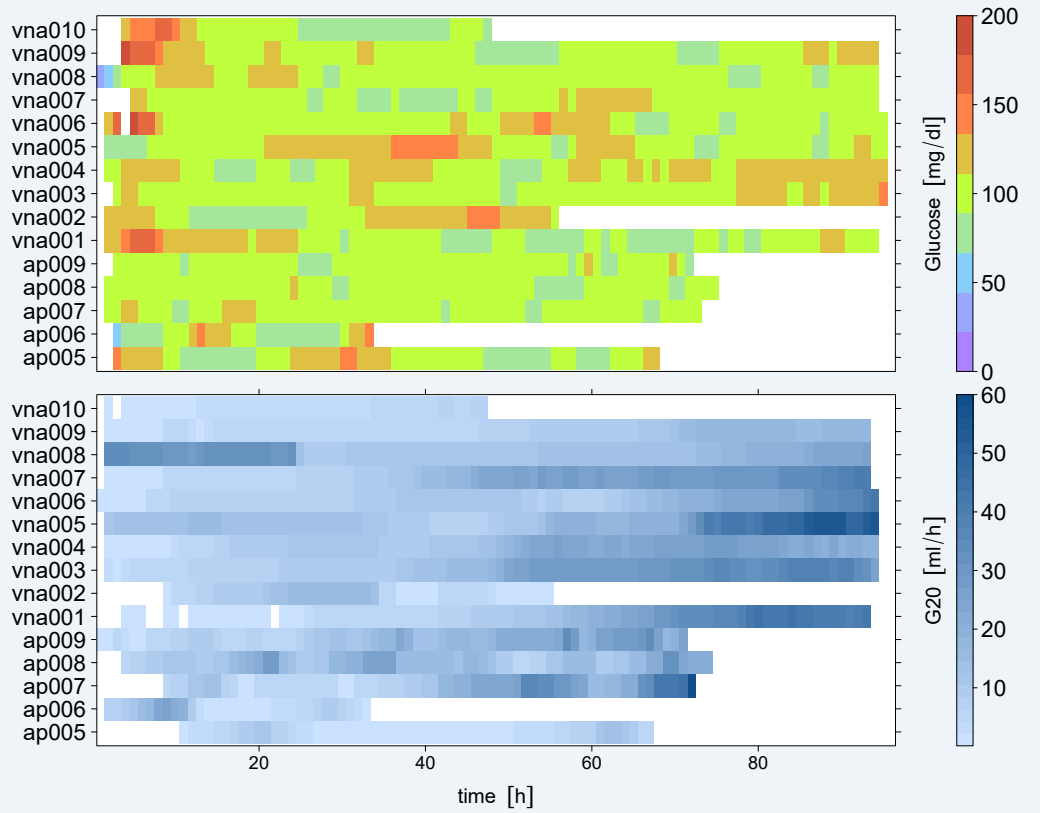


Figure 13.11: Overview of one hour means of the measured glucose levels (mg/dL) for all automated trials. If no measurement was performed, the last known measurement is carried forward.

13.4.2.2 Other controlled parameters

In addition to the glucose management, limited evaluations for the other parameters have been performed, if they were required for the individual animal subjects.

13.4.2.2.1 Base excess Management of base excess by adaption of a sodium bicarbonate (NaBic) infusion pump was performed for 5 animal subjects. The results of this management are shown in Table 13.4. Base excess could be kept within a physiological range of −2–2 mmol/L in 53 % of the 289 observed hours. Measurements < -4 mmol/L were observed for 62 hours and significantly increased levels >4 mmol/L for 6 hours. A significant deviation from the average performance can be seen for animal subject vna003, where the desired range could only be kept for 8% of the observation time. An overview of the measured levels and pump rates for all animal subjects is shown in Figure 13.13. Additionally, an example for the dynamic adaption of the infusion rate performed by the fuzzy logic controller for management of base excess in animal subject ap009 is shown in Figure 13.12.

Table 13.4: Results for control of base excess with a target of 0 mmol/L.

	base excess ranges [mmol/l]						
	≤−4	(−4,−2]	(−2,2]	(2,4]	>4	hours	within (−2,2]
ap007	0	1	60	4	6	71	0.85
ap008	0	1	57	15	0	73	0.78
ap009	11	7	39	12	0	69	0.57
vna003	33	53	7	0	0	93	0.08
vna004	18	19	44	2	0	83	0.59
total	62	81	207	33	6	389	**0.53**

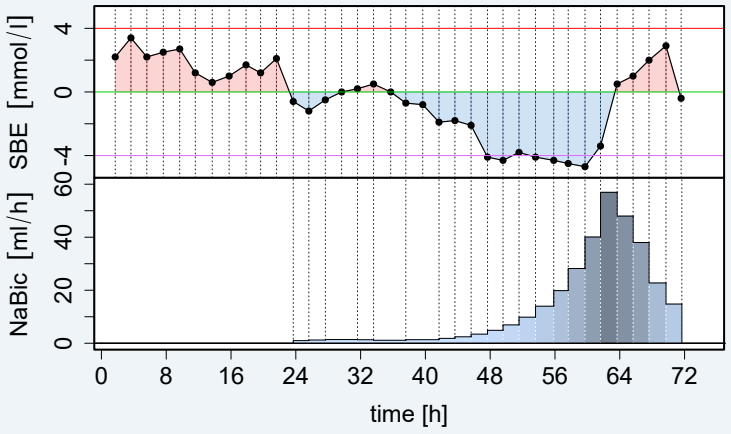


Figure 13.12: Example for NaBic control depending on base-excess level for subject ap009.

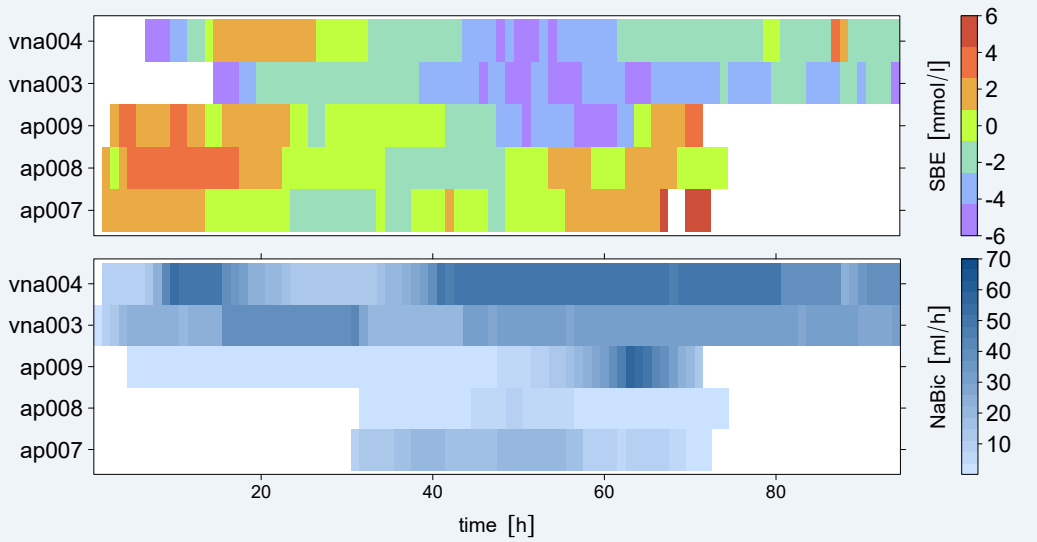


Figure 13.13: Base excess: Overview of one hour means of the measured levels for all automated trials. If no measurement was performed, the last known measurement is carried forward.

13.4.2.2.2 Potassium Further evaluation of the fuzzy controller was performed by implementing management of potassium levels by adaption of a potassium chloride (KCl) pump for 7 animal subjects of the VNA study. The achieved performance of this management is shown in Table 13.5 and Figure 13.14. During the observation time of 567 hours, physiological levels (3.5–4.5 mmol/L) could be maintained for only 44 % of the time. Yet, severe hypo- and hyperkalemia were not observed in any animal subject.

Table 13.5: Results for potassium management with a target level of 1 mmol/L.

	potassium ranges [mmol/l]						
	≤2	(2, 3.5]	(3.5,4.5]	(4.5,6]	>6	hours	within (3.5,4.5]
vna003	0	58	34	0	0	92	0.37
vna004	0	47	41	3	0	91	0.45
vna005	0	56	31	6	0	93	0.33
vna006	0	18	62	13	0	93	0.67
vna007	0	1	18	5	0	24	0.75
vna008	0	61	26	6	0	93	0.28
vna009	0	33	42	15	0	90	0.47
total	0	274	254	48	0	576	**0.44**

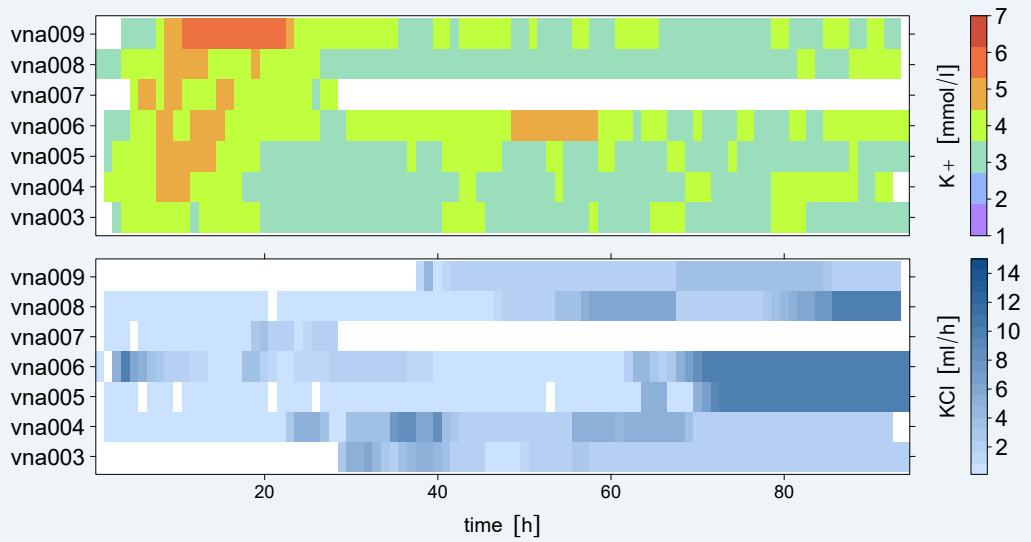


Figure 13.14: Potassium: Overview of one hour means of the measured levels for all animal subjects. If no measurement was performed, the last known measurement is carried forward.

13.4.2.2.3 Calcium Calcium management was performed in 5 animal subjects of the AP study. The results of this management are shown in Table 13.6. A physiological level within 0.8–1.2 mmol/L could be maintained for 213 of 313 observed hours (68 %). Hypocalcemia with levels ≤0.6 mmol/L was observed for 4 hours. Significant hypercalcemia was not observed at any time. An overview of the measurement results and pump rates is shown in Figure 13.15.

Table 13.6: Results for calcium management with a target level of 4 mmol/L.

	calcium ranges [mmol/l]						
	≤0.6	(0.6,0.8]	(0.8,1.2]	(1.2,1.6]	>1.6	hours	within (0.8,1.2]
ap005	0	23	41	2	0	66	0.62
ap006	3	11	18	0	0	32	0.56
ap007	1	16	47	7	0	71	0.66
ap008	0	9	64	1	0	74	0.86
ap009	0	20	43	7	0	70	0.61
total	4	79	213	17	0	313	**0.68**

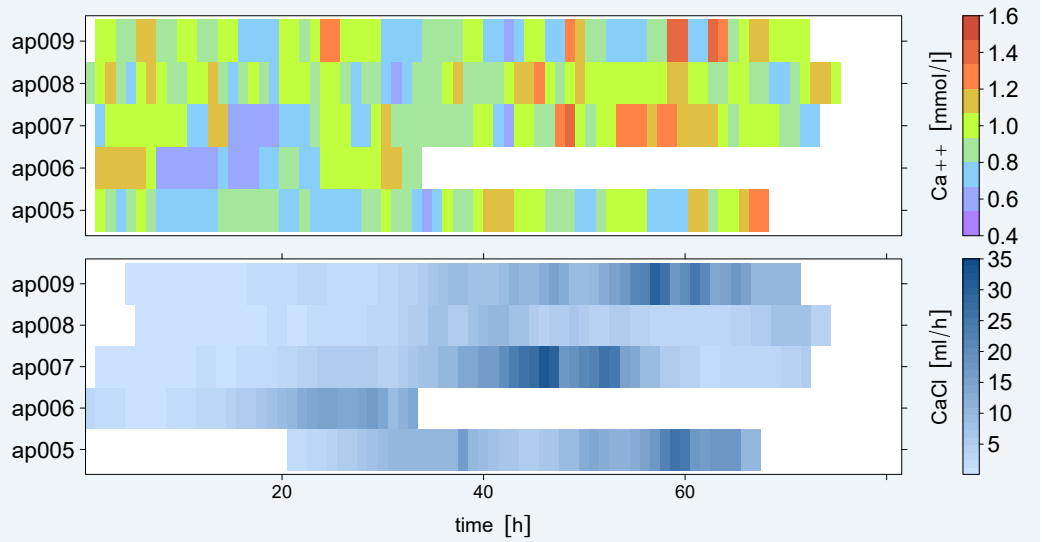


Figure 13.15: Calcium: Overview of one hour means of the measured levels for all automated trials. If no measurement was performed, the last known measurement is carried forward.

13.5 Discussion

For application of closed-loops in medical decision making and the used experimental study setting, a closed-loop system based on a fuzzy-logic controller was developed and applied. By using fuzzy logic controllers, clinical expertise can easily be integrated as it can use linguistic descriptions and be derived from therapeutic protocols which are often of fuzzy nature without sharp numerical boundaries [7]. Thus, fuzzy logic is a concept successfully applied to various different tasks in medical decision making [116, 137–140, 174, 179].
Using the proposed controller, inputs and outputs can easily be configured to be any parameter transmitted via message objects within the TICoMS. This allows for rapid evaluation of novel concepts as well as easy adaption and expansion of existing controllers. In comparison to other fuzzy-logic based approaches, the proposed fuzzy-logic controller uses only relative output factors to control the medical devices. By usage of such a factor, no adaption of the output function to the individual numeric ranges of the controlled variable needs to be performed. Instead, the output can directly be applied to a variety of tasks. Due to these relative adaptions of the current setting, a goal-directed therapy by annealing to a desired target level is performed. Manual therapeutic interaction and adaptions of the current settings are possible at any time. After such manual interventions, the target levels can be adapted, otherwise the controller will just start a stepwise return to the previously set goal. Furthermore, the controller can be started and stopped at any point in time, as the only internal and absolute state are the most recent measurements for slope calculation and averaging in the next processing step. Those can simply be omitted in a first processing step after (re-)start. Using those relative factors together with the message-based output, a decoupling from a specific controlled device is achieved as the output is simply a message,

distributed by the server to any application with matching filter rules. For example, if an infusion pump has a malfunction, caregivers can easily remove the defective pump and set up another pump with the appropriate drug and an approximately matched infusion rate. The system will then continue the goal-directed therapy by applying the rate changes to this replacement pump. Apart from such a failure scenario, this interchangeability and the possibility of intervention at any given state may be important features in inter- or intra-hospital transfer, where patients may arrive with already running infusion pumps that need to be integrated in the control system without any further change of pumps or configurations.

Evaluation of the proposed goal-directed closed-loop fuzzy-logic controller was successfully performed during the AP and VNA study in the used porcine ICU setting for management of $etCO_2$, glucose, base excess, potassium, and calcium to maintain homeostasis and provide stable conditions for the observations. Further parameters were only evaluated briefly for several hours or individual animal subjects and not included in this evaluation as no reliable evaluation can be performed on the limited data base. For the developed and presented closed-loop controls, the individual needs of the animal subjects can clearly be observed with the collected measurements and resulting infusion rates.

The proposed closed-loop control for RR was successfully applied to maintain a constant $etCO_2$ level. Due to the relationship between exhaled CO_2 and the CO_2 concentration in the blood, the acid-base balance could be stabilized as an essential step for maintaining general homeostasis. For almost all animal subjects, the desired CO_2 level could be achieved. In three animal subjects (ap005, ap007, vna003), an increased target level (45 mmHg) was used as medical conditions or sensor offsets required an adaption. For one animal subject, a problem with $etCO_2$ sensor of the ventilator arose, resulting in a measurement offset. Therefore, the fuzzy set had to be adapted for compensation of the wrong measurement. This was easily performed by adaption of the desired target level and shows the importance of a flexible system that can be adapted by the caregiver at any time to counter technical problems and provide adequate care if medical conditions indicate adapted target levels during therapy.

For respiratory management, the implemented locking function for the fuzzy set proved to be a vitally important feature during the performed medical procedures and maneuvers by freezing the current state and prohibiting changes during those measurement episodes. Especially during the VNA evaluation, where the RR is purposely set to 0 for a short period of time as a respiratory hold, aggregation of measurements would have been significantly skewed. Regarding the aggregation time itself, the shorter time frame used in the VNA study allowed for a tighter control with better results. Admittedly, the prolonged post-operative observation time with more stable conditions and the study protocol with less common interruptions as well as the larger sample size, might contribute to this bias.

Glucose management was successfully performed using the proposed controller. Due bedding times of up to 96 hours in the VNA trials without other means of nutrition, the G20 infusion was an essential energy source and thus the most important application of the de-

veloped fuzzy logic controller. Hypoglycemic events below 70 mg/dL were only observed in post-operative animal subjects at the beginning of the observation and not caused by the control algorithm. On contrary, hyperglycemia was present for some post-operative animal subjects. As no insulin was administered, no direct intervention could be performed and it took some time for the animal subjects' blood glucose levels to descend to nominal levels. Once this range was reached, management was successfully applied. Other approaches as proposed by Thompson et al. observed severe hypoglycemic events in 1.42 percent of the 12.886 performed glucose measurements [150], a single hypoglycemia event with a measured glucose level below 40 mg/dL was observed by Pachler et al. [152] during their study with 60 patients observed for 72 hours each. The observations by Kulnik et al. [154] included 10 patients with an average observation time of 66 hours where glucose levels between 80 mg/dL–110 mg/dL could be achieved in 47 % of the observation time. In comparison to these results, the developed control system showed a comparable or better performance and provided a fully automated solution. Of course, for direct comparison between the results, all study conditions would need to be identical and differences in the patient demographic, their nutrition, illness and other factors influencing the glucose levels would need to be known and considered.
Management of base excess by intravenous infusion of buffering NaBic solution was successfully evaluated for 5 animal subjects. Yet, a significantly decreased performance was observed for animal subject vna003. Since ventilation was difficult for this animal subject, the target etCO_2 level was increased from 40 mmHg to 45 mmHg. Of course, this increase influenced the gas exchange and intravascular CO_2 concentration. Such increased levels lead to an acidification of the blood, which is most likely reflected in the decreased base excess. By excluding this animal subject with the previous reasoning, successful management could be performed in 68% of the time for the remaining 4 animal subjects.

The next managed parameter, potassium, could only be kept within the desired range in 44% of the total observation time and significant differences regarding the performance, ranging from 28% to 75% for the individual animal subjects, are observed. Since the fuzzy control for potassium was only evaluated for the VNA study, no comparison to more stable conditions in the AP study can be performed. However, given the aim of this study, investigating the volume state of the animal subjects, different defined states of volume deficit and excess were evaluated by administering of diuretics and supplementary volume, respectively. This caused significant changes in intravascular fluid content, as a result of which the dilution of the solved potassium cations varied. Additionally, the different usage of diuretics for the individual animal subjects may have further contributed to the observed variations. Because of these dependencies, factors such as infusion rates and diuresis should be included in the fuzzy controller for further evaluation.

Calcium management by the proposed fuzzy controller was performed for 5 animal subjects of the AP study. Levels were kept within the desired target range for 68% of the total observation time. Whereas this application showed the successful implementation of the fuzzy controller, the need for this control was not indicated for further animal subjects. Retrospective analysis of the obtained BGAs of the VNA study showed that only 5 hypocalcemic

events occurred in a total of 932 observed hours and the nominal level could be maintained without closed-loop control for around 70% of the time. This leads to the conclusion that the proposed controller does not have a significant impact on the calcium levels, which are successfully regulated by the body's internal closed-loops under the given study conditions. Further studies where this buffering ability of the body may be impaired, e.g., kidney failures or alkalosis, would be required to perform an assessment of benefits for such an automated calcium management.
Of course, this general fuzzy controller required simplifications compared to other more specific and model-based approaches. Yet, for clinical application and easy configuration, a simple input is required and no system with more parameters that need to be decided on by the physician can be put to practical use. Additionally, model-based approaches require large sets of training data or expertise and are tailored to a single, specific parameter. As the developed approach was specifically designed for the experimental research environment where the easy application and evaluation of novel closed-loops was desired, no such prior knowledge and models should be required.

Regarding the measurement frequency, BGAs were performed every few hours to achieve a tight control. This high frequency is required for a dynamic system and remains a major drawback and is not limited to the proposed controller but a general problem of related closed-loop systems, requiring measurements to be obtained nearly every hour [147, 152]. The benefit of high measurement frequencies could clearly be shown for the controller applied to management of $etCO_2$, where such a high data frequency was available. Another complication is the manual process of blood withdrawal for BGA. A subsequent measurement might be carried out in half an hour or not within the next 8 hours. Further complications are measurement inaccuracies and waiting times for the analysis to be performed as processing of other samples or regularly executed maintenance programs on the blood gas analyzer may delay the sample processing. Therefore, the regulatory closed-loop is fully dependent on this manual task and a mere dependency on blood gases should be avoided if possible. Novel sensors for continuous glucose monitoring (CGM) may provide a reliable and high data rate, yet available sensors are expensive or still in an experimental state and mostly tailored for long-term implantation during diabetes management [148, 157]. Thus, not applicable in critical care as a disposable product. For this reason, currently further indicators and surrogate parameters should be considered.

A further comparison to closed-loop systems used in clinical trials shows that such approaches are always based on individual laptops or integrated in isolated devices with no direct access to other measurements and patient data. Thus, the measured glucose levels and other parameters are required to be entered manually. On contrary, the developed fuzzy logic controller integrated in the unified TICoMS is able to obtain the required measurements automatically, reducing workload and human error potential. In addition, the required changes are directly applied to the infusion pumps, providing further reduction of workload and allowing a faster response. Of course, given the current legal framework, the control part can only be carried out in an experimental research setting, as automated therapy guidance in clinical trials is very restricted and requires manual confirmations by the caregivers.

Considering further safety aspects, it is important to care about various safety features in automated systems to prohibit decisions that may cause patient harm. Therefore, as shown for the implemented pump and fuzzy logic controllers, as a final step before applying changes to medical devices, limiting checks for maximum rates and settings should always be performed. Given the evaluated examples, conditions like low respiration rates causing oxygen depletion or excessive infusion of possibly lethal drugs, like KCl, were avoided by those limitations. Furthermore, manual interaction should remain possible at any time and have immediate effect, as complicated overriding procedures may cause or prolong life-threatening conditions.

13.5.1 Outlook

With the current implementation as a proof of concept for using a fuzzy logic controller within the developed software framework, further steps may include the improvement and expansion of the current system.

Whereas the implemented fuzzy sets were used as test cases for the controller and the developed framework, the complexity of those closed-loops was still limited. More complicated interaction of different parameters, e.g., total infused volume in relation to the dilution of the administered medications may be included to provide a more adaptive and precise system. Given the current consistent rule base, expansion to multiple parameters can easily be performed. Additional fuzzy functions for pairs of absolute values and slope of a second parameter could be integrated to reflect further interactions and dependencies for a controlled parameter. The used weighting between absolute value and slope could be expanded to an additional weighting between the multiple input parameters and their slopes.

Regarding the data collection and control of the devices, the most important aspect would be an improved connection to the infusion pumps. Due to the limited and experimental remote-control interface provided by the manufacturer, data rate and collected information were still limited. Especially for highly sensitive and critical infusions like Arterenol, infusion pumps with a faster response time and more information about current pump state, including the remaining infusion fluid for administering a bolus, would be required. Given the current implementation, pumps could no longer be controlled if they entered a warning state, e.g., when the amount of infusion fluid gets low and syringes need to be replaced, which may lead to critical situations if not recognized.

For scenarios with very asymmetrical risks, the distribution of the member functions within the fuzzy sets might not be optimal, as proportional factors derived from the three used input values (target, lower citical, and upper critical levels) are used. Extremely skewed settings thus might cause member functions to overlap to a high or low degree, impairing the control algorithm. Hence, such extreme scenarios may need to be evaluated to improve the performance of the controller.

For respiratory management, sensor offsets and measurement inaccuracies may influence control. Automatic recalibration of the target value based on BGA may be considered, as in some cases an offset was present and in one case (vna003) the target had to be artificially increased until sensor recalibration and replacement could be performed. Of course, such offsets may not only be of technical nature, as some medical conditions can disturb

the correlation between blood and expiratory CO_2 levels. Yet, such technical difficulties showed the single point of failure. To prohibit such problems and sole dependencies, surrogate parameters or redundancy by sensor fusion with additional sensors for improved reliability would be useful. Additionally, further control of metabolic processes and consideration of additional parameters like temperature and cardiac parameters might improve CO_2 management.
Beside specific improvements for individual control applications, the controller itself might be improved by a more general and variable description. Recently, a standardized fuzzy makeup language was proposed as the IEEE standard 1855-2016 for modeling of fuzzy systems [202]. Modeling of the controller according to this novel standard could provide an additional abstraction layer between the controller, integrated into the medical framework, and the control logic. Thus, implementation and evaluation of different controllers with varying complexities and enhancements may be performed more easily and by definition in an additional configuration file. This might allow for dynamic adaptions of the controller during run-time by utilizing machine-learning approaches.

Yet another important aspect for clinical application is the need for a more adaptive and interactive configuration of the fuzzy sets by the physician as shown for sensor offsets or clinical conditions that required adaptions of the therapeutic goals. Whereas the used XML-based configuration files provide all necessary settings, adaptions can only be performed by a quick edit of this configuration file once the fuzzy controller is stopped. An enhanced version may utilize the central patient monitor and the touchscreen interface to perform such adaptions during run-time. This may include the assignment of new fuzzy sets and adaption of existing ones using a GUI. Moreover, it may provide warnings to the caregiver if critical conditions are observed. A concept for such an integral fuzzy control system is shown in Figure 13.16.

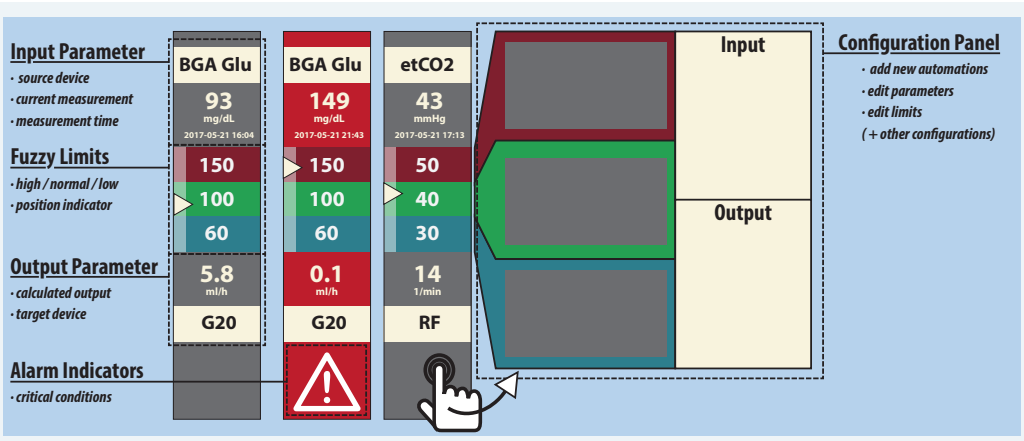


Figure 13.16: Design draft for a fuzzy GUI integrated in the control touchscreen for overview and control of multiple vital parameters.

Part V

Out of the Loop: Data Analysis and Prediction

Prediction is very difficult, especially about the future.
old Danish proverb

14 Data classification and pre-processing

14.1 Motivation

Using the developed framework with the implemented closed-loop systems, clinical data has been collected in a homogenous database for further analysis. Within the clinical setting and the given study conditions, specific devices and configurations have varied between individual subjects or were not used for intermediate time frames depending on the therapeutic needs. Furthermore, given the experimental research setting, the configuration of the devices was changed several times in the course of the studies and occasional hot-fixes as well as updates sometimes led to short down times during data collection.
Thus, as many machine learning approaches rely on a fixed vector of input parameters, a first assessment regarding the availability and quality of the over 300 collected parameters in the raw data base had to be made prior to further processing.
This assessment must not be limited to the availability of data, because even if a measurement result is collected, it might be artificial or erroneous. Such an observation can have various causes, including inaccurate internal calculations by the medical devices or manual interventions interfering with the correct measurement. An example for the first case are irregularities or intermediate heart beats due to a bad signal to noise ratio and measurement artifacts due to breathing. Regarding to the second cause, pressure readings and other sensors might be impaired by accidental touching and disturbance of the setup or intentional therapeutic actions.
Therefore, prior to any further analysis and application of machine learning approaches, a general overview of the hundreds of collected parameters should be obtained. This offers important information on data availability, frequencies, and characteristic patterns that may facilitate further analysis by distinguishing reliable results from data that should be excluded from further steps or used only for tasks other than such an analysis.

14.2 Method

Using the developed export pipeline with the created HDF5 files for each animal subject, the collected data was imported to Matlab for further analysis. Since this analysis should be applicable for all collected parameters, a general approach, not requiring specific models and knowledge for over 300 individual parameters, was chosen. Such an enhanced and in-dept processing may only be performed for selected parameters in further steps. In order to obtain additional information beyond mere availability, an automatic classification of each data point was performed. This classification provided information on available, measurement delays, changes in measurement frequency, and mean deviation.

Analysis is carried out independently for each individual parameter and animal subject. Given the complete measurement series, a sliding window approach with window size c_t is used to evaluate each point depending on the previous measurements. This subset is the analysis vector $v = (x_{t_0-c_t}, \ldots, x_{t_0})$, containing the current data point $v(c_t)$ and the chosen subset of previous observations.
With this data frame, evaluation of the current observation is performed in several steps regarding three different scopes: the absolute relation of the current value in comparison to the earlier observations, its relative change, and changes in the frequency of obtained measurements.
In a first step, general statistics, including mean $\hat{v}$, median $\widetilde{v}$, standard deviation s, and two quantile boundaries q_1 and q_2 are calculated for the observation vector v. Based on those boundaries, a nominal data range $[\hat{v}-s, \hat{v}+s]$ is defined.

For the next steps, a filtered time series v_{nom}, containing only measurements within $[\hat{v}-s, \hat{v}+s]$ is used to reduce the effect of outliers. Given enough data points ($|v_{nom}| > obs_{min}$), a robust regression using iteratively reweighed least squares with a bi-square weighting function is performed to obtain an estimator v'_{nom} for the observations within v_{nom}. For computation in Matlab the implemented `robustfit` with default bi-square settings is used. This function utilizes a tuned calculation of the regression factors for handling large residuals with adapted standard deviation $s = MAD/0.6745$ and a tuning factor $f = 4.685$ for the calculation of the residuals $r = r_{prior}/(f \cdot s \cdot \sqrt{(1-h)})$.
Once this regression is calculated, the quality is assessed by calculation of the absolute error $err = abs(v_{nom} - v'_{nom})$. This estimator for the goodness of fit is used with and additional slack variable c_{slack} to define a numerical reference range $range_{nom} = err \cdot c_{slack}$ for classification of nominal values.
Next, the temporal dimension is analyzed. This is conducted by calculating the time differences t_{diff} between each two successive measurements within v and estimating the median data rate $\widetilde{t_{diff}}$. Additionally, utilizing the calculated time differences between successive observations from the previous step, the slopes are calculated. Those are averaged to obtain a median relative rate of change $\widetilde{\Delta}$ for comparison to the current observation.
Given those statistical measures for the vector v with the length c_t in the numerical and temporal domain, the last step consists of the classification of the current observation $x = v(c_t)$ with respect to those metrics, resulting in a binary vector containing flags for: (1) data available; (2) absolute thresholds exceeded ($x \notin [\widetilde{v} - range_{nom}, \widetilde{v} - range_{nom}]$) ; (3) data late ($(t_{diff}(v_t) - t_{diff}(i)) > c_{late} \cdot \widetilde{t_{diff}}$, where i is the previous available measurement in v); and (4) relative rate change greater than threshold ($\Delta v_t, v_{t-1} > \widetilde{\Delta}$).
An overview of this processing pipeline is shown in Figure 14.1. Analysis is repeated for all parameters and successive time points with a shifted observation vector v.

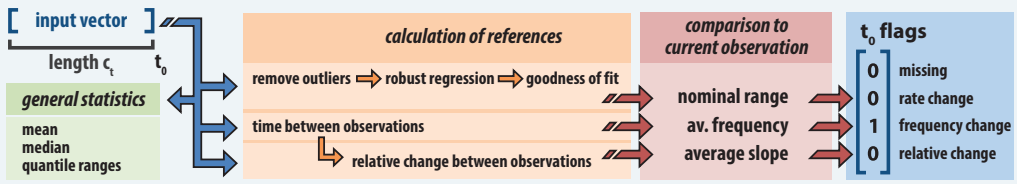


Figure 14.1: Schematic overview of the general analysis pipeline for all observed parameters. Different binary flags for the characteristics of the currently observed data point at t_0 are calculated by using a vector of previous observations.

14.3 Application to collected study data

For application of this classification pipeline to the collected study data, an observation time $c_t = 3600$ with a window size of one hour was used. Regression analysis was performed if more than 20 measurements were present ($obs_{min} = 20$), otherwise only a comparison to the median observation was performed. Quantile boundaries $q_1 = 0.2$ and $q_2 = 0.8$ were used and the nominal data range was defined using $c_{slack} = 1.1$. For classification within the time domain, a delay tolerance of $c_{late} = 1.5$ was used. A special analysis case was the processing of the obtained BGA measurements. Since most of the time only one measurement was available in the observation window c_t, just the availability flags could be calculated.

14.4 Results

Using the described analysis pipeline, automated classification for each observation of the studies was performed to obtain an overview of the data distribution and availability. Examples for the classification for different parameters and periods of time are shown in Figure 14.2. Successful classification of missing data points and deviations can clearly be observed in all examples. Since analysis was performed for all parameters and time points, it was possible to generate overview graphics including all parameters of the animal subject using R. Such a summary is shown in Figure 14.3 for animal subject ap010. The analysis results are presented as one-minute averages of the availability flags. Devices like ventilator and scales with the highest resolution of 1 Hz are shown in black. Lower data rates are colored gray. The different start times of the medical devices can clearly be recognized by the individual start of the recording. Outages are displayed in red and, depending on the cause, occurred for individual parameters, single devices, or device groups. At the end of the study, the shutdown of the devices can clearly be observed. Apart from this retrospective analysis, the classifier was successfully evaluated as a real-time application with a data rate of 1 Hz.

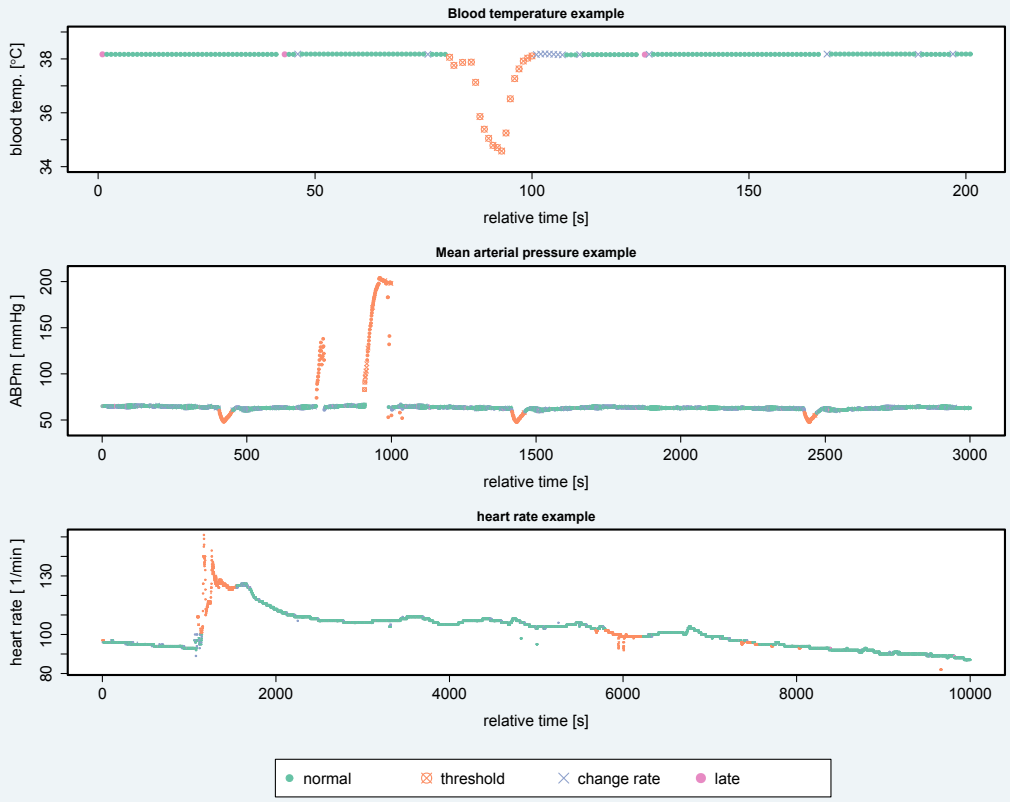


Figure 14.2: Examples for point analysis and classification or detection of outliers and absolute and temporal changes with different analyzed and plotted timeframes for blood temperature during injection of cold fluid (top), mean arterial pressure with artifacts (middle) and heart rate (bottom). The flags, represented with colored symbols, show the nominal states and if a deviation is detected by exceeding the dynamically calculated threshold, change rate, or maximal timeframe between measurements.

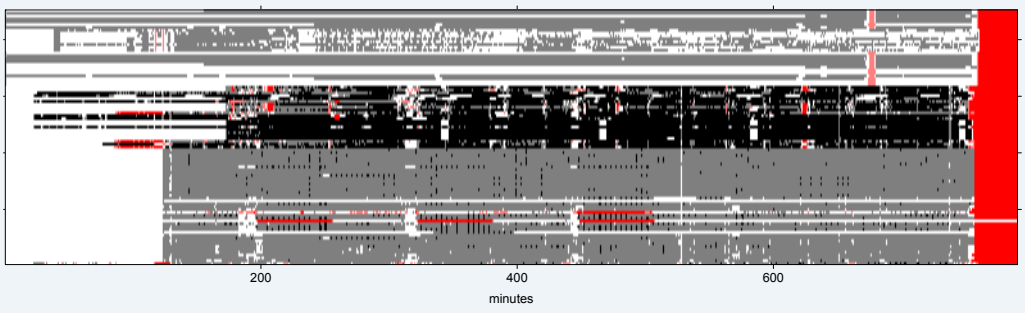


Figure 14.3: Overview of the point analysis for all parameters of animal subject ap010 as one minute averages. Missing data is shown in red and shades of gray indicate the average data frequency.

14.5 Discussion

The presented method is an intermediate step to gain a first insight to the collected data. The algorithm uses a sliding window to calculate nominal ranges for each observation to detect outliers and provide a dynamic, but robust reference for the quality of each observation. This provides additional information for the collected data that can be used in further processing steps, including but not limited to the detection of outliers. By using the sliding window, considering the most recent observations, dynamic adaption of the boundaries in the course of time is performed. To account for the uncertainty of the performed regression, the goodness of fit is evaluated and used as a slack factor for the classification. Whereas mainly used retrospectively for processing of the collected data, a real-time version was tested as well and might be used to provide a general overview of the various parameters with a fast detection of potential problems in real-time.
The used approach was especially designed for general application to all parameters with different numerical ranges and characteristics to detect significant events like sensor drop-offs, bad pressure reading, or other errors in a dynamic ensemble of medical devices and therapeutic interactions. Such events should be detected in collected data to reduce or eliminate the influence of artificial observations in further analysis. As only an intermediate step to guide additional processing and analysis, no further evaluation of the performance and comparison to other methods was conducted. Using output flags as supplementary information, differentiation between errors in individual sensors, entire devices, or the framework for data collection itself may be performed. In comparison, most other methods are designed for data imputation and do not calculate additional flags, that might be processed further.
Such further analysis may include the detection of different states, as parameters might be NMAR if they share a common cause. Especially the combination of different parameters and specific patterns of outages or delays may be useful if it originates from specific therapeutic procedures. One such application may be an automated segmentation of the observations into different phases during operation or critical care. Once given a general overview regarding specific problems and requirements for individual parameters, further analysis pipelines and processing may then be used to develop more specific and improved models for individual parameter sets. In anticipation of the next chapter, such characteristic patterns could be observed in the overviews of the analyzed animal subjects for the arterial pressure curves during blood withdrawals. This observation, a combination of artificial pressures and data NMAR, was then used as a detection algorithm for those events.
On the other hand, device malfunctions, accidental disconnects, or other events may cause data to be MCAR without specific patterns or patterns that show individual devices as malfunctioning. This can beneficial for the development of smart alarms and notification of caregivers. Yet, given the current implementation, the dynamic adaption of the uncertainty range may not be optimal for this problem as large uncertainties without fixed boundaries can limit the detection of outliers. For the main goal, obtaining an overview of the collected data prior to further processing, this dynamic adaption provided no problems.

15 Analysis and prediction of BGA results

As shown in results for the closed-loop applications, limited temporal resolution and availability of BGA results was a major obstacle and uncertainty factor for automated control and management of homeostasis. Thus, the ability to interpolate or predict BGA measurements for intermediate control or guidance to achieve shorter adaption intervals was found to be the most important aspect for improvement of the developed closed-loop system. Given the general overview of the collected data presented in the previous chapter, the next steps to approach this goal were obtaining matched training datasets of BGA- and surrogate parameters and a first evaluation of the performance for such a prediction. As processing of blood samples can induce a significant and variable delay of several minutes between monitoring and sample data, determination of the withdrawal timepoint was essential for this task and of general interest in clinical practice.

15.1 Detection of blood withdrawals and other manipulations

15.1.1 Motivation

As BGA is a laboratory measurement involving manual processing steps and possible delays, further analysis for developing a prediction model and subsequent application for automated predictions and recalibration may only be performed if the blood sample can be matched to other collected parameters. Therefore, an automated detection of blood withdrawals and other manipulations is an essential step to provide the required data and detect measurement artifacts.
Additionally, detection of withdrawals is of clinical relevance as blood sampling is one of the essential tasks in critical care, which has to be carried out manually every few hours or, if clinically indicated, even more often. One of the most common errors during handling of blood samples is patient misidentification [203]. Various precautions such as checklists and bar-codes are used to ensure proper sample dating and patient identification as critical and significant steps for an adequate patient care [203–205]. Yet, errors, observed in up to 1 per 1000 samplings [206–208], may have life-threatening consequences if critical conditions are not recognized or assigned to the wrong patient. However, those steps still require manual work and may even provide additional workload, which means that noncompliance with these guidelines continues to pose a major problem [203, 209, 210]. It is therefore a purposeful and forward-looking strategy to solve this problems with technical methods that ensure reliable identification [211, 212].
To perform this crucial task, a detection algorithm was developed and published [213]. The concept of this approach will be presented in the following with an extended evaluation for additional animal subjects.

15.1.2 Blood sampling process

In order to develop a technical solution for withdrawal detection, it is important to understand the sampling process and how it affects patient monitoring. Blood withdrawal from an arterial line with established arterial pressure monitoring is in general performed by an interruption of pressure monitoring to obtain access. This procedure is illustrated in Figure 15.1. Such interventions cause unavoidable artifacts and characteristic patterns, e.g., artificial blood pressure and no pulsation, in the monitoring data that provide difficulties for further processing and need to be filtered for further analysis [130, 214]. Such patterns were also found in the general analysis of the recorded data and formed the starting point for recognition. An example for such a typical withdrawal pattern is shown in Figure 15.2. The observed mean arterial pressure (ABPm) reaches an artificial high level, whereas systolic (ABPs) and diastolic (ABPd) pressures cannot be calculated due to the lack of pressure variation and thus are not transmitted by the patient monitor.
While these artifacts pose problems for the further processing of the data, they can be used for the detection of blood withdrawals and for validation of blood samples with recently detected withdrawals from a specific patient. In addition, it provides an essential tool for matching blood samples to other data aiding further analysis and machine learning.

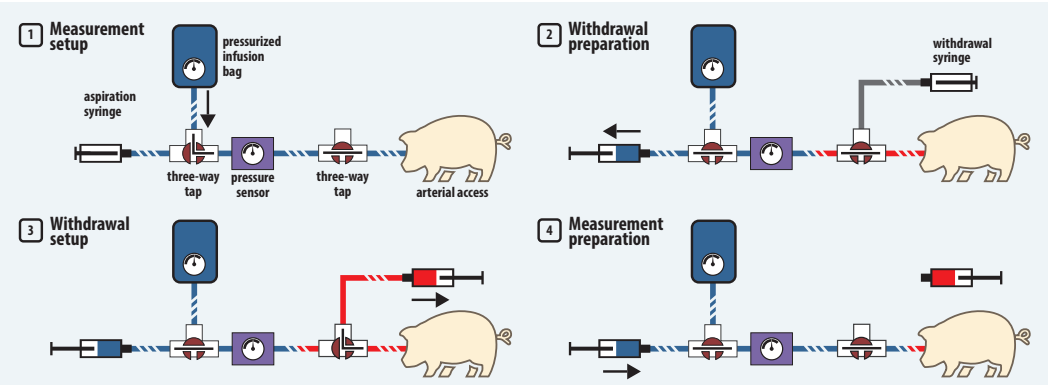


Figure 15.1: Schematic illustration of the arterial pressure measurement setup and interruption during arterial blood withdrawal. During measurement (1) a standing fluid pillar is detecting pressure changes. For withdrawal, arterial blood is aspirated (2) and withdrawn with a syringe (3). To continue measurement, the aspirated arterial blood is re-injected through the arterial access (4) and pressure measurement is continued.

15.1.3 Detection algorithm

With the help of the basic observations regarding patterns in the monitoring data during blood sampling, a detection algorithm for blood withdrawals was developed [213]. This algorithm uses available and missing information derived from the classification of the arterial pressure monitoring data to detect the exact time of blood withdrawal. However, detection is not limited to those specific case but can be used for general detection of events based on a variable number of input parameters and calculated flags. Thus, first a general description is given and then applied to the concrete withdrawal detection problem.

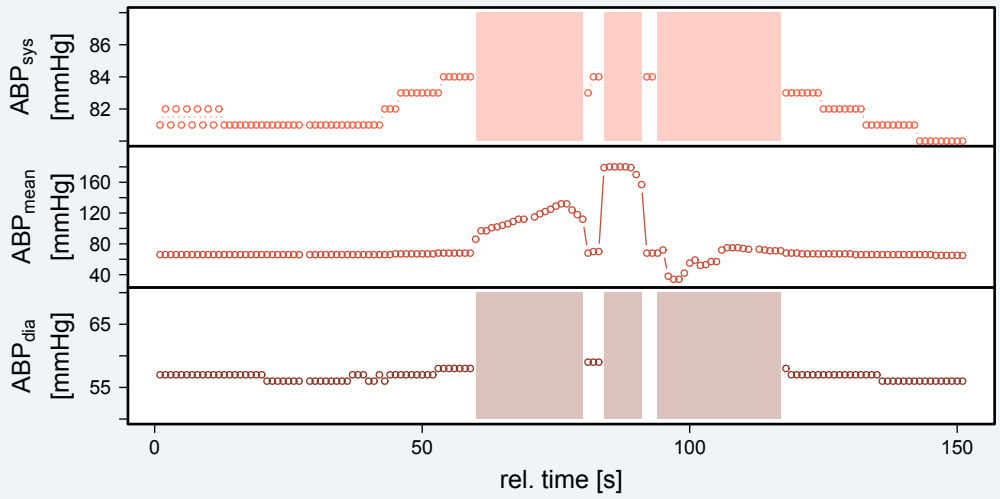


Figure 15.2: Example observation of arterial pressure (ABP) readings from the patient monitor during a blood withdrawal for animal subject ap006. Significant variation of the arterial mean pressure and lack of calculations (highlighted with boxes) for systolic and diastolic calculations by the patient monitor can clearly be observed.

In a first step, individual binary scores $s_i \in [0,1]$ for the defined input parameters are calculated. These scores are based on the various available flags of the data classification which were described in the previous section. Additionally, each parameter is associated with a individual weight w_i for contribution to a total score S. To calculate the individual scores, the obtained measurement of a parameter i is processed once a second to obtain an individual score s_i for this parameter. If multiple observations were detected within a single second, the most recent measurement is used for further processing.
For the individual scores $(s_1, \ldots s_n)$, the total score $S = \sum_{i=1}^{n} w_i \cdot s_i$ with $\sum_{i=1}^{n} w_i = 1$, resulting in a normalized scoring function within the interval $[0,1]$, is calculated. The interval boundaries of the scoring function $s = 1$ and $s = 0$ indicate that either all indicators for a blood withdrawal are present or no indication for such an event was given, respectively.
Based on this score s, a simple threshold S_{Th} is used to determine if a blood withdrawal event was present when $S > S_{Th}$. To obtain a more robust result and avoid false detections for cases where only a single point in time exceeds the threshold, a series of S_n successive points in time must reach the threshold S_{Th} to qualify as a blood withdrawal event.

15.1.3.1 Detection of manipulations at the arterial access

Detection of manipulations and withdrawal events at the arterial access with the described algorithm is performed by utilizing the ABPm, ABPs and ABPd pressure readings for calculation of the scoring function. For ABPm, the score s_{ABPm} is based on the detection of deviations from the mean observation of the last 10 minutes, ignoring missing data. If a deviation is more than 10 mmHg from the mean the individual score s_1 is 1, otherwise 0. For ABPs and ABPd, binary flags for data availability are used ($s_i = 1$ if a measurement is present).

Based on the individual scoring functions for ABPm, ABPs, and ABPd, the combined score as a weighted combination is used as an indicator for withdrawal events. The total score S is calculated from the weighted average ($w_1 = w_2 = w_3$) of the individual scores for all three parameters. The threshold for the scoring function is set to $S_{Th} = 0.7$ and manipulation events are reported if this threshold is exceeded for 10 successive seconds ($S_n = 10$).

15.1.4 Evaluation

Evaluation of the proposed algorithm was performed as a retrospective analysis for all animal subjects of the performed AP and VNA studies where data collection from the patient monitor was available. Due to the short observation time and various required therapeutic interventions influencing the pressure measurement, animal subject ap010 was excluded from this analysis. The arterial pressure information was obtained from the Philips MP50 patient monitor with TICoMS during the studies and stored in the central database. Withdrawals were performed by physicians, scientists, medical students, and lab technicians, thus providing a broad variety of individuals and skill levels. For retrospective analysis, the collected information was extracted from the study database and each point in time was processed in successive order, simulating the online behavior of the algorithm.
Each detected event was automatically plotted using a MATLAB script to provide a graphic chart covering a 10-minute window. This was used to perform a visual inspection and validation of the variables and the scoring function for each event. As the used blood gas analyzer (Radiometer ABL800 FLEX) was connected to the developed system as well, the analysis results and processing times were stored with the monitoring information in the central database. For evaluation of the withdrawals, those events were exported and processed in the same manner as the automated detected events to be used as a reference.
Each automatically detected event then was matched to the reference analysis times if an assignment was possible and counted as *hits* or *misses*, respectively. For additionally detected events, manual visual inspection and classification of legitimate withdrawals or flushing events with clearly visible artificial pressure levels (as *others*) and false positive detections (as *FP*) was performed. Finally, using this classification of all detected and known withdrawal events, performance of the detection was evaluated in terms of sensitivity, precision, and F1 score.
In addition to the retrospective analysis, a proof of feasibility for the real-time version of the algorithm was performed for two animal subjects. As a reference, blood withdrawals were observed manually and the resulting deviations in ABPm, ABPd, and ABPs on the IntelliVue MP50 monitor were compared to the automatic detection by the proposed algorithm.

15.1.5 Results

Using the proposed evaluation method, the performed BGAs were matched to the detected withdrawal events. All additional events were evaluated and classified using the generated plots. A detailed example of such a detected event with the plotted measurements and scoring function is shown in Figure 15.3. Using this analysis for all detected events, the algorithmic performance for each animal subject and as a whole was evaluated and is pre-

sented in Table 15.1. A total number of 886 BGAs was carried out during the performed studies. The proposed algorithm detected 1198 events of which 866 were the performed BGAs. Thus, BGAs could be successfully detected in 866 of 886 cases (97.74%). In addition, 268 further events, including blood withdrawal for laboratory and ACT analysis, or flushings of the arterial lines, were correctly detected. In 47 cases, the algorithm performed a false positive event detection, which, easily recognizable by visual inspection of the plotted data, was based solely on fluctuations in the pressure curve. Overall, a sensitivity of 0.98, a precision of 0.96, and an F1 score of 0.97 were achieved. An overview of all detected of the individual animal subjects matched to the performed BGAs is shown in Figure 15.4. As observable in this figure, nearly all BGAs (97.3 %) were successfully detected in the AP study with regular withdrawal patterns and the number of additionally detected events was very small (2.9%). For the VNA study, most BGAs were successfully matched as well. However, numerous additional detection events (30%) can be observed. In this study, BGAs were performed at defined changes in the total body volume, resulting in dynamic and variable sampling intervals that can clearly be observed.
A real-time version of the algorithm was successfully integrated in the developed software framework and was able to successfully detect blood withdrawals and manipulations from the arterial catheter in real-time for two animal subjects.

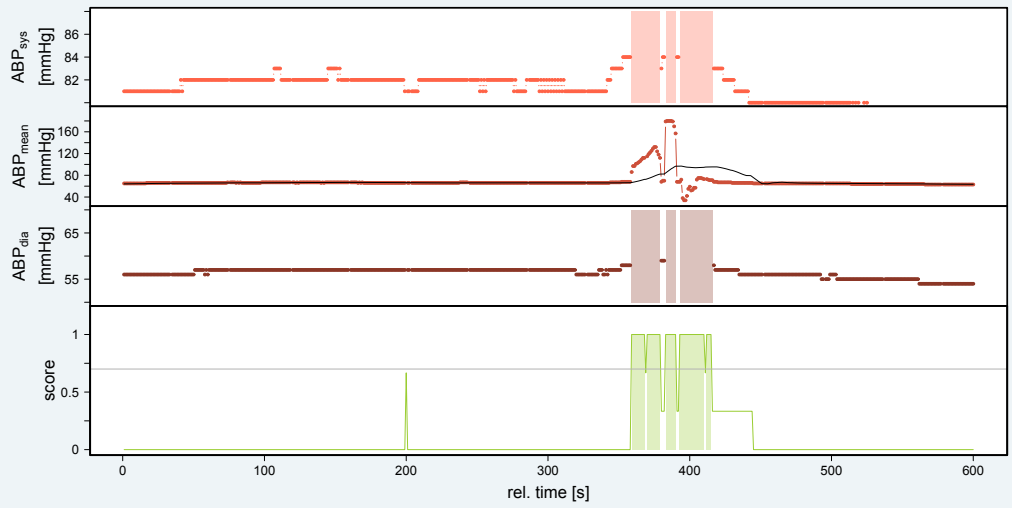


Figure 15.3: Detection example with the proposed scoring for the same event as in the motivation for animal subject ap006.

15.1.6 Discussion

The detection of blood withdrawals using pressure monitoring was tailored to a usual blood withdrawal process from an arterial access. Whereas those events lead to unavoidable artifacts in pressure measurements that need to be filtered for further analytical processing of the pressure curve [130, 214], they provide essential markers for automated detection of blood withdrawals and interactions. Weighting and combining individual parameters as

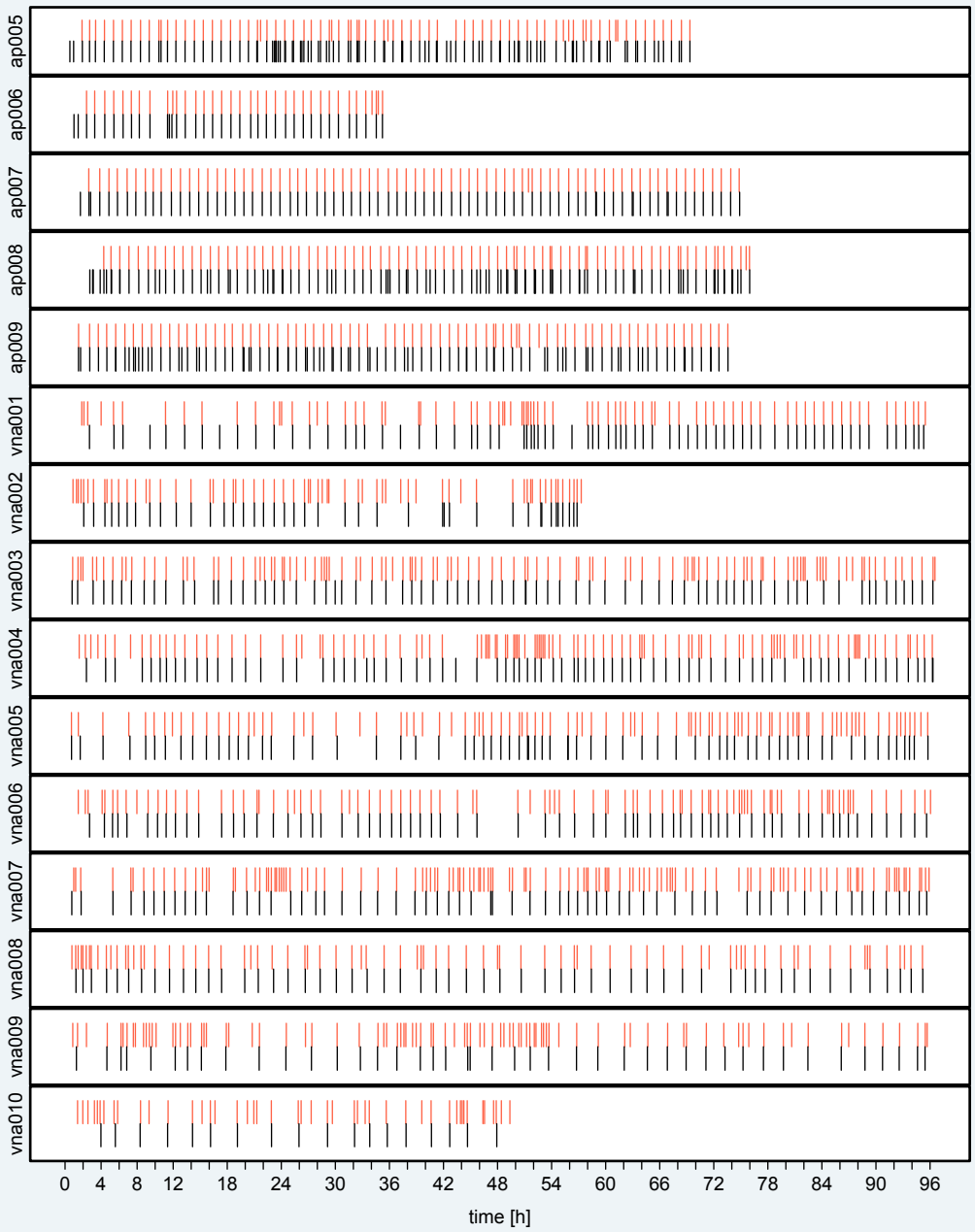


Figure 15.4: Overview of all detected events (red) and the BGA references (black) for all analyzed animal subjects. During the AP study, stable conditions with fixed BGA intervals were maintained. The VNA study was performed with purposeful states of volume deficits, causing difficulties maintaining arterial access which required aspirations and flushings that can be observed as additional interaction events.

Table 15.1: Performance for the individual animal subjects. Presented are the number of performed BGAs, the number of detected events, the number of correctly identified BGAs (hits), not detected BGAs (misses), other withdrawals and flushings (other), false positives (errors) and applicable scores. (Extended from [213])

	VNA									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
#BGAs	66	40	74	73	60	67	62	54	42	17
#manip.	78	57	109	108	94	89	119	77	88	40
hits	66	38	73	68	59	65	62	54	42	17
misses	0	2	1	5	1	2	0	0	0	0
other	8	15	15	35	30	22	47	20	44	22
errors	4	4	4	5	5	2	10	3	2	1
sen.	1.00	0.96	0.99	0.95	0.99	0.98	1.00	1.00	1.00	1.00
prec.	0.95	0.93	0.96	0.95	0.95	0.98	0.92	0.96	0.98	0.98
F1	0.97	0.95	0.97	0.95	0.97	0.98	0.96	0.98	0.99	0.99

	AP				
	5	**6**	**7**	**8**	**9**
#BGAs	72	34	75	77	73
#manip.	77	36	74	79	73
hits	69	33	73	77	70
misses	3	1	2	0	3
other	5	2	1	1	1
errors	3	1	0	1	2
sen.	0.96	0.97	0.97	1.00	0.96
prec.	0.96	0.97	1.00	0.99	0.97
F1	0.96	0.97	0.99	0.99	0.97

AP & VNA		
Summary		
#BGAs	Σ	886
#manip.	Σ	1198
hits	Σ	866
misses	Σ	20
other	Σ	268
errors	Σ	47
sen.		0.98
prec.		0.96
F1		0.97

algorithmic input allows an easy adaption and expansion of the algorithm depending on the available data and needs. For example, the blood temperature could easily be added to detect flushing events. However, since arterial temperature measurement using a PiCCO catheter system is not always available, this additional indicator was omitted for general application and during the performed evaluation.

In this evaluation, a maximum permissible time between sample collection and analysis had to be assumed, as the detected events had to be matched to the performed analysis. If a BGA was not detected within the chosen 10-minute window, it was counted as a miss. This scenario may have happened in some cases where required medical interactions or cleaning programs of the blood gas analyzer delayed processing of the blood sample. Thus, no interactions could be observed within the pressure curves for the reported analysis time but only more than 10 minutes prior. Accounting for those events, further improvement of performance may be achieved, as the presented values may be seen as a lower boundary.

For the AP study, with animal subjects observed in a stable condition with fixed BGA intervals of 2 hours, a one to one matching of manipulations and BGA events could be achieved

in most cases. On contrary, in the VNA study, evaluating different volume states that included severe volume deficit, multiple additional events were detected. Such extreme states made management of the arterial access more difficult, frequently requiring aspirations and flushings of the arterial access to maintain monitoring. Those therapeutic interactions were of course detected as well and different volume states may even be classified by the changes in the frequency of manipulation during the course of observation.
In order to be used as a detection algorithm for withdrawals and validation of blood samples, it is important to recognize each performed withdrawal as undetected events cannot be dated and compared. Thus, the sensitivity of the algorithm is more important than specificity. Whereas other events and manipulations, as seen in VNA study, may lead to false positives, the number of such additional events is still limited and did not cause problems for the matching of individual BGAs.
Another important aspect is variability in the blood withdrawal process as it requires several steps to be performed manually, possibly interfering with the algorithms performance. During the performed studies with the animal subjects, interactions and blood withdrawals were performed by a broad variety of caregivers with different levels of professional experience and backgrounds. This resulted in variations of the blood withdrawal process, including the individual steps and required amount of time. The developed algorithm was able to successfully detect those events for all caregivers, showing feasibility for general application in a clinical environment and allowing a precise dating of blood samples by correcting the time offset added by sample collection and analysis.

15.1.6.1 Outlook

As patient misidentification [203] is one of the most common errors in handling blood samples, detected withdrawal events and times may be used as a reference for validating blood samples originating from a specific patient. This may provide an improvement for blood gas analyzers, as the patient selection can be limited to patients where a manipulation was recently detected. As correct dating and matching of blood samples is essential for adequate patient care [203–205], such an automated solution can increase safety as it becomes effective automatically in the pre-analytical phase [212], whereas guidelines and other approaches still rely on the human compliance [203, 209, 210] to avoid life-threatening situations due to mix-ups [211]. Even further enhancement of patient safety may be performed with automated validation of the BGA results with other vital parameters.
Regarding the detection method of the algorithm, further improvement may be achieved by including additional parameters or a more specific scoring flags like duration of the spikes or the numerical sign of the changes to further differentiate between flushing and withdrawals. If a large enough training data set is available, this might be performed as a supervised or unsupervised classification task utilizing machine-learning approaches.
Despite the evaluation with different caregivers and critical states of volume deficit, further evaluation regarding to various patient conditions would still be necessary. Since the algorithm was evaluated on animal subjects under general anesthesia, no voluntary movement and other artifacts were present in the collected data. For clinical application, especially patient movement may cause interruptions and significantly decrease performance and thus should be considered and evaluated.

15.2 BGA prediction with surrogate parameters

15.2.1 Motivation

As BGA may only be performed irregularly every few hours, difficulties for closed-loop control due to the low data frequency can arise. Additionally, BGAs are invasive measurements associated with infection risk, cost of disposables, and high manual workload for the caregivers during sample collection and processing. Thus, finding surrogate parameter with a higher temporal resolution can be a helpful solution and will aid the development of predictive models for BGA parameters, allowing to establish tighter closed-loop control and better homeostasis management. And even if this goal of an accurate prediction cannot fully be achieved, a detection of significant changes observed in the surrogate parameters will still be very beneficial to indicate the imminent need of a BGA to the caregiver.
Another important aspect, as described in the previous section, is the matching of blood samples to the correct patient for patient safety. Because check-lists and other approaches to avoid sample mix-ups still have an associated human risk, technical solutions becoming effective in the pre-analytical phase may be beneficial for patient identification and avoidance of sampling mix-ups [211, 212]. If blood gases can be derived from other vital parameters, this prediction may be used for validation of the blood sample.

Independently of the goal to be achieved, the most important aspect for any analysis was to obtain matched pairs of monitored vital signs and BGA results. Unfortunately, the process of blood sampling requires some time for the withdrawal process itself, sample transportation, and analysis. It may even further be delayed if the blood gas analyzer is busy or not ready for processing of the sample. This can easily result in varying delays of several minutes during which vital parameters that can be correlated to the blood sample may change. Whereas for simple correlations between closed-loop systems and the measured parameters, an offset of a few minutes may be neglected, for machine learning, using a vector of many, even a few hundred, input parameters, the relations between those values and the BGA results are important. Especially, when considering the experimental setting with different analytical maneuvers, a stable state is disturbed with those interactions and various changes as part of the analytical protocol may be required. For example, in the VNA study, the respiratory hold maneuver and Trendelenburg leg-raise test were performed directly after blood withdrawal and could drastically change various vital parameters, resulting in a significant mismatch to BGA results.
Given the developed method for automatic detection of withdrawal events described in the previous section, BGA results can be matched to withdrawal events and thus to other observed vital signs, solving the delay problem as additionally illustrated in Figure 15.5. Using this essential foundation of a matched dataset, surrogate parameters that are readily available from patient monitoring may now be used to improve the closed-loops or patient identification. Yet, for finding such surrogate parameters and a first evaluation, several steps still had to be performed. Given the raw, matched dataset, the first required step was generating a training dataset that can be used for further processing and machine learning. Then, using the obtained dataset, prediction of blood gas parameters with different

approaches and evaluation of the achieved performance could be conducted. In what follows, those steps will be presented.

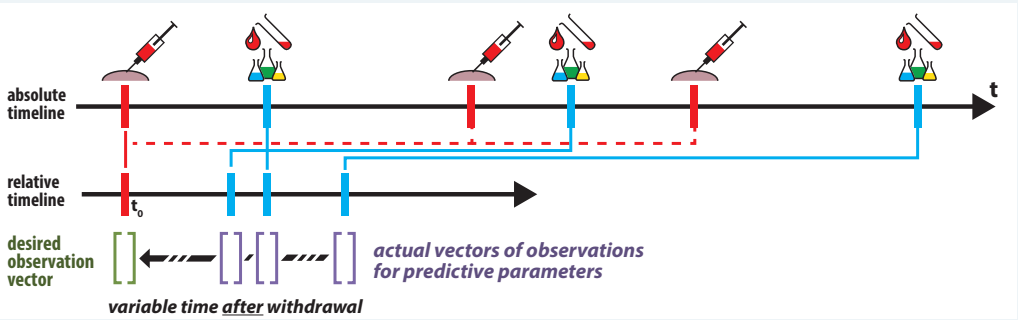


Figure 15.5: Visualization of parameter offset resulting from withdrawal and analysis delay. Predictive values are obtained from various points in time after the withdrawal if no correct dating of the withdrawal is performed.

15.2.2 Methods

15.2.2.1 Creation and evaluation of the training dataset

In a first step for machine-learning-based predictions of blood gas surrogate parameters, a training dataset had to be generated. Using the automated detection of blood withdrawal events to get the specific withdrawal times, a raw dataset containing the BGA results and matched monitoring data without a measurement offset for each performed BGA of the VNA study (animal subjects vna001 – vna010) was obtained. Within this dataset, containing the union of all parameters observed during the study, missing observations for individual measurements or parameters in individual animal subjects were still included. Yet, for further processing with machine-learning approaches, a complete training dataset without such gaps was required. To obtain this dataset, a subset had to be calculated. Therefore, a tradeoff between the removal of entire unreliable parameters and incomplete observations had to be found. This was performed by solving the maximization problem for remaining data points in relation to a variation of the threshold for removal of an entire parameter. After finding this threshold and reduction of the dataset, a subset of parameters relevant to the BGA prediction task was chosen, excluding alarm limits and other technical parameters.

15.2.2.2 Prediction of blood gas parameters

Using the calculated training dataset, prediction of surrogate parameters could be performed with a broad variety of machine learning approaches that require complete input vectors. Yet, due to different numerical ranges of the vital parameters, a scaling of the training data was needed to avoid a prediction bias toward larger numerical parameters. This problem was solved by scaling each parameter by the 90% quantile of its absolute values.
For analysis of this now scaled dataset with 14 different blood gas parameters, three regression methods were tested. First, ridge regression was used to provide a base-line for regression performance. As parameter reduction was a desired goal, lasso regression with these capabilities was tested as well. To gain an insight how parameter reduction influences

the result, a third analysis using elastic-net regression, a weighted combination of the L1 and L2 norms used by lasso and ridge, was evaluated for the entire range of α weighting factors. For each regression method, models for each of the 14 parameters were trained and evaluated using leave-one-out (n-1) cross-fold validation. Training was performed on 9 animal subjects of the VNA study and performance was tested on the remaining one in each step.

The λ parameter for each of the 14 models within this 10-fold cross-validation was automatically selected by using another nested 10-fold cross-validation on the training dataset. The optimal value for λ within each cross-fold then was selected from a geometric sequence of 100 values, with the largest just sufficient to produce $\beta = 0$. Performance of each regression model was assessed by calculation of MSE, NRMSD, and R^2 for the test subject of each cross-fold, excluded from training. The overall performance was then calculated as the average performance for all cross-folds.

For the third performed analysis using elastic-net regression, 100 different α values between 0 and 1 for steps of 0.01 were evaluated in the same way. In fact, as ridge and lasso are only edge cases of elastic-net regression with $\alpha = 0$ and $\alpha = 1$, respectively, those two edge cases were already covered. To assess the influence, the results for all α parameters were collected. The average performance for each value of α was calculated as the average of the models obtained from the 10 cross-folds. The best α for each BGA parameter was then chosen as the one yielding the smallest average MSE.

Given the best α value, the achieved parameter reduction was observed. To provide an estimate how many parameters have a significant influence and if parameter reduction is reasonable for the evaluated BGA parameter, the average number parameters required to contribute to 50 percent of the regression coefficients was calculated and is denoted as β_{50}. For this calculation, the regression coefficients were sorted in descending order by their absolute value and the total contribution of the parameters was accumulated until a sum of 0.5 was exceeded.

Previously, only the MSE was used as a measure for the goodness of fit. Yet, often relative changes and correlations are more important for parameters in the clinical field. To provide an estimate of how the selection of an optimal α for the elastic-net regression influences this performance, the previously described steps for selection of the best α were performed with the goal of selecting the minimal coefficient of determination (R^2) as well. A schematic overview of the nested regression pipeline and selection of optimal α values for the evaluated BGA parameters, based on the average model performance of the cross-fold validation, is shown in Figure 15.6.

Finally, besides evaluation of elastic-net regression and assessment of optimal α parameters for each of the examined BGA parameters, the results are briefly compared to SVR. This evaluation was performed in a similar fashion, using the scaled input data and by performing an n-1 cross-fold validation.

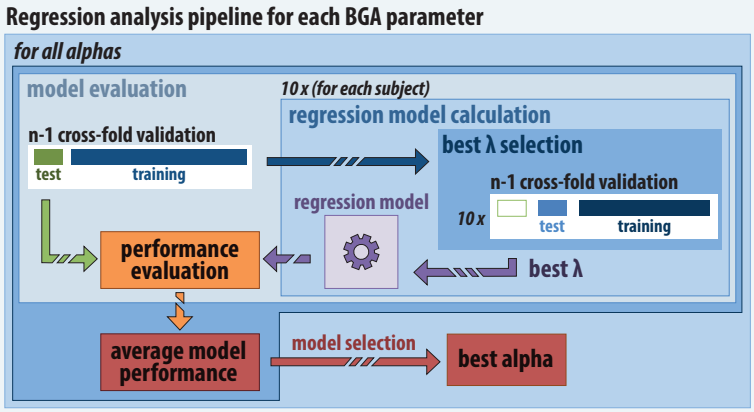


Figure 15.6: Schematic illustration of the processing pipeline. Elastic-net models were evaluated for different α values using 10-fold cross-validation with nested cross-validation for λ selection. Based on the best average performance in the trained models, α was selected.

15.2.3 Results

15.2.3.1 Optimization of training dataset

In a first step, the raw dataset was preprocessed to obtain a reduced dataset suitable for training. For the VNA study, an optimal threshold of 38 missing observations for exclusion of a parameter was calculated and is visualized in Figure 15.7. Omitting technical parameters and alarms, 498 BGA observations of 88 parameters from patient monitor, ventilator, infusion pumps, Pulsion PulsioFlex, Osypka ICON, body, and diuresis scale remained.

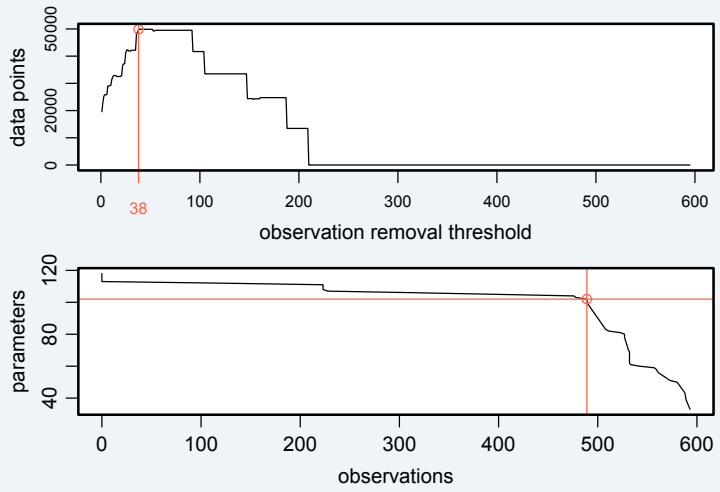


Figure 15.7: Optimization of the training dataset by maximizing remaining data points (top) and the relation between remaining observations and parameters (bottom). If a parameter was unavailable in more than 38 observations, it was excluded.

15.2.3.2 Regression analysis

Using the training data set obtained in the previous step, scaling of each parameter was carried out by its 90% quantile. In this way, training bias towards individual parameters was avoided and possible outliers were taken into account, which would also have led to a bias if normalization had been performed on the basis of the largest observed value.

Next, regression analysis was performed. The results for all performed regressions for the evaluated range of α parameters, including ridge and lasso regression as edge cases, are shown in Figure 15.8. For each BGA, the calculated MSE of each cross-fold, the average of all, and the optimal alpha are visualized on top of the results regarding the r^2 performance below. The results of this evaluation are summarized in Table 15.2 for the three used methods (ridge, lasso and elastic-net). For elastic-net regression, the best α, based on the average MSE of the cross-fold models, is denoted. During the evaluation of the performed lasso regression of cCa^{2+}, no r^2 could be calculated as the prediction yielded a constant term with zero variance. For most evaluated BGA parameters, ridge regression or elastic-net regression with small values of α achieved the best results. Yet, the influence of α for most parameters, as observed in Figure 15.8, is very small. Parameter reduction yielded the best performance for $FCOHb$ ($\alpha = 0.98$), $FMetHb$ ($\alpha = 0.69$), and temperature ($\alpha = 1.00$).

By observation of the individual cross-fold models in Figure 15.8, similar performance, with an exception for the models evaluated on animal subject vna002, can be seen for most parameters. For this subject, only a limited set of test cases could be evaluated as the observation time was cut short due to medical complications. For SBE and T, a significant variation between the individual cross-fold models can be observed.

Using the chosen optimal α parameters for each examined BGA parameter, the true values for the test cases of each cross-fold can be plotted against the predictions as shown in Figure 15.9. For visualization of the data distribution, hexplots with color-coded density are used. It can be observed that a majority of obtained results could be predicted with a high accuracy, except for $COHb$ and $MetHb$, which show a bifurcation in two clusters of observations for individual animal subjects. Yet, for the remaining parameters, the high accuracy is mostly limited to a cluster around a single point with high density. With increased distance from this center of the distribution, a decreasing accuracy is observed.

15.2.3.3 Selection regarding the best coefficient of determination

Additionally to the optimization of the α value in regard to a minimal MSE of the cross-fold test cases, a model selection based on a maximized coefficient of determination R^2 was performed to assess the influence of this goal regarding the choice of α. The model selection regarding this goal is included in Figure 15.8 and shown below calculated MSE of each parameter, respectively. The results of this optimization and the best α parameters are presented in Table 15.3.

Comparison of the individual cross-folds, presented in Figure 15.8, show a similar performance for most cross-folds. The chosen value of α does not have a significant influence on the achieved performance, except for temperature T, where parameter reduction leads to a significant increase.

Table 15.2: Evaluation results for regression analyses and MSE optimizations (rounded to 3 digits). The average number of coefficients contributing to at least 50 percent of the β coefficients is denoted as β_{50}.

		$FCOHb$	$cCa^{2}+$	$cGlu$	cK^{+}	$cLac$	$FMetHb$	cNa^{+}
ridge	MSE	2.416	0.037	314.527	0.440	2.238	1.020	176.718
$\alpha = 0$	NRMSD	-3.245	0.194	0.162	0.159	0.914	0.941	0.077
	R^2	0.056	0.037	0.079	0.110	0.261	0.051	0.190
lasso	MSE	2.537	0.048	687.543	0.841	8.657	1.469	533.715
$\alpha = 1$	NRMSD	-5.008	0.213	0.229	0.207	1.758	1.030	0.120
	R^2	0.052	–	0.117	0.081	0.243	0.053	0.389
optimum	α	0.98	0.01	0.02	0.00	0.00	0.69	0.00
elastic net	MSE	2.352	0.036	293.691	0.440	2.238	0.901	176.718
	NRMSD	-4.951	0.192	0.159	0.159	0.914	0.885	0.077
	R^2	0.058	0.056	0.156	0.110	0.261	0.074	0.190
	β_{50}	2.3	7.4	19.4	14.8	18.1	2.1	14.2

		FO_2Hb	SBE	T	pCO_2T	pHT	pO_2T	$ctHb$
	MSE	36.596	14.146	0.572	33.129	0.004	1192.076	1.175
$\alpha = 0$	NRMSD	0.044	-0.756	0.018	0.120	0.008	0.236	0.106
	R^2	0.040	0.118	0.049	0.080	0.112	0.063	0.261
lasso	MSE	43.441	10.994	0.397	59.069	0.010	4211.333	2.643
$\alpha = 1$	NRMSD	0.054	-0.708	0.014	0.159	0.012	0.428	0.145
	R^2	0.048	0.116	0.610	0.079	0.053	0.283	0.242
optimum	α	0.00	0.06	1.00	0.33	0.00	0.00	0.00
elastic net	MSE	36.596	10.391	0.397	31.577	0.004	1192.076	1.175
	NRMSD	0.044	-0.690	0.014	0.119	0.008	0.236	0.106
	R^2	0.040	0.105	0.610	0.088	0.112	0.063	0.261
	β_{50}	13.4	16.6	1	8.5	20.3	11.5	15.3

15.2.3.4 Optimal model parameters

Based on the optimized α values regarding MSE and R^2, predictive parameters may be derived from the influence of the individual regression coefficients. Therefore, the normalized regression coefficients were sorted by their absolute weight in descending order. The weight distribution of the individual regression coefficients for the optimal α for each predicted BGA parameter are shown in Figure 15.10. The difference between the weights for optimal MSE and R^2 of each coefficient is highlighted with a thick bar on a smaller stem for each parameter. Whereas for some parameters a broad distribution with small regression coefficients for numerous parameters can be observed, especially for temperature T, $FCOHb$, cCa^{2+}, $FMetHb$, and pCO_2T a small fraction of significantly contributing coefficients can be recognized. This can be observed in Table 15.2 as well, where those parameters have a large optimal α and a small β_{50} value. For BGA parameters with significant contributions of individual regression coefficients, contributing parameters with their relative coefficient weight are shown in Table 15.5. Comparing the two evaluated measures for the goodness of

Table 15.3: Evaluation results for the regression analyses and optimizations regarding the goodness of correlation based on R^2 (rounded to 3 digits). The average number of coefficients contributing to at least 50 percent of the β coefficients is denoted as β_{50}.

		$FCOHb$	cCa^2+	$cGlu$	cK^+	$cLac$	$FMetHb$	cNa^+
optimum	α	0.72	0.31	0.01	0.01	0.04	0.58	0.94
elastic net	MSE	2.432	0.046	302.428	0.478	2.329	0.952	573.355
	NRMSD	-4.824	0.211	0.161	0.167	1.108	0.909	0.118
	R^2	0.063	0.110	0.159	0.111	0.322	0.076	0.467
	β_{50}	2.7	1.1	19.6	12.5	15.3	2.4	3.5

		FO_2Hb	SBE	T	pCO_2T	pHT	pO_2T	$ctHb$
optimum	α	0.92	0.27	1.00	0.95	0.04	0.97	0.23
elastic net	MSE	40.357	12.828	0.397	54.437	0.011	2311.109	2.958
	NRMSD	0.049	-0.745	0.014	0.153	0.012	0.336	0.149
	R^2	0.049	0.154	0.610	0.112	0.164	0.289	0.311
	β_{50}	3.7	13.3	1	3.3	8.3	5.0	9.5

fit, MSE and R^2, only small variations in the regression coefficients for the most significant contributing parameters can be observed. The only notable exception is cCa^{2+}, where 9 factors are contributing to 50% of the regression coefficients in MSE optimization, yet regarding R^2 the only significantly contributing factor is tissue perfusion.

15.2.3.5 Comparison to SVR

For comparison, an additional evaluation using SVR, using a linear kernel and similar n-1 cross-fold validations for the 10 animal subjects of the VNA study, was performed. Evaluation results of this model are presented in Table 15.4. Neither this linear kernel nor additional tests with other kernels provided significantly better results for MSE and R^2 than the optimized elastic-net approach.

Table 15.4: Averaged results for the test cases of the 10-fold cross-validation using a SVR approach.

		$FCOHb$	cCa^2+	$cGlu$	cK^+	$cLac$	$FMetHb$	cNa^+
SVR	MSE	4.512	0.264	350.641	1.912	2.378	1.672	381.429
linear	NRMSD	-3.346	0.431	0.173	0.292	0.910	1.104	0.100
	R^2	0.025	0.067	0.175	0.084	0.302	0.039	0.385

		FO_2Hb	SBE	T	pCO_2T	pHT	pO_2T	$ctHb$
SVR	MSE	50.978	12.939	1.074	54.303	0.010	1301.463	4.134
linear	NRMSD	0.055	-0.774	0.023	0.154	0.012	0.260	0.175
	R^2	0.047	0.060	0.299	0.043	0.073	0.217	0.255

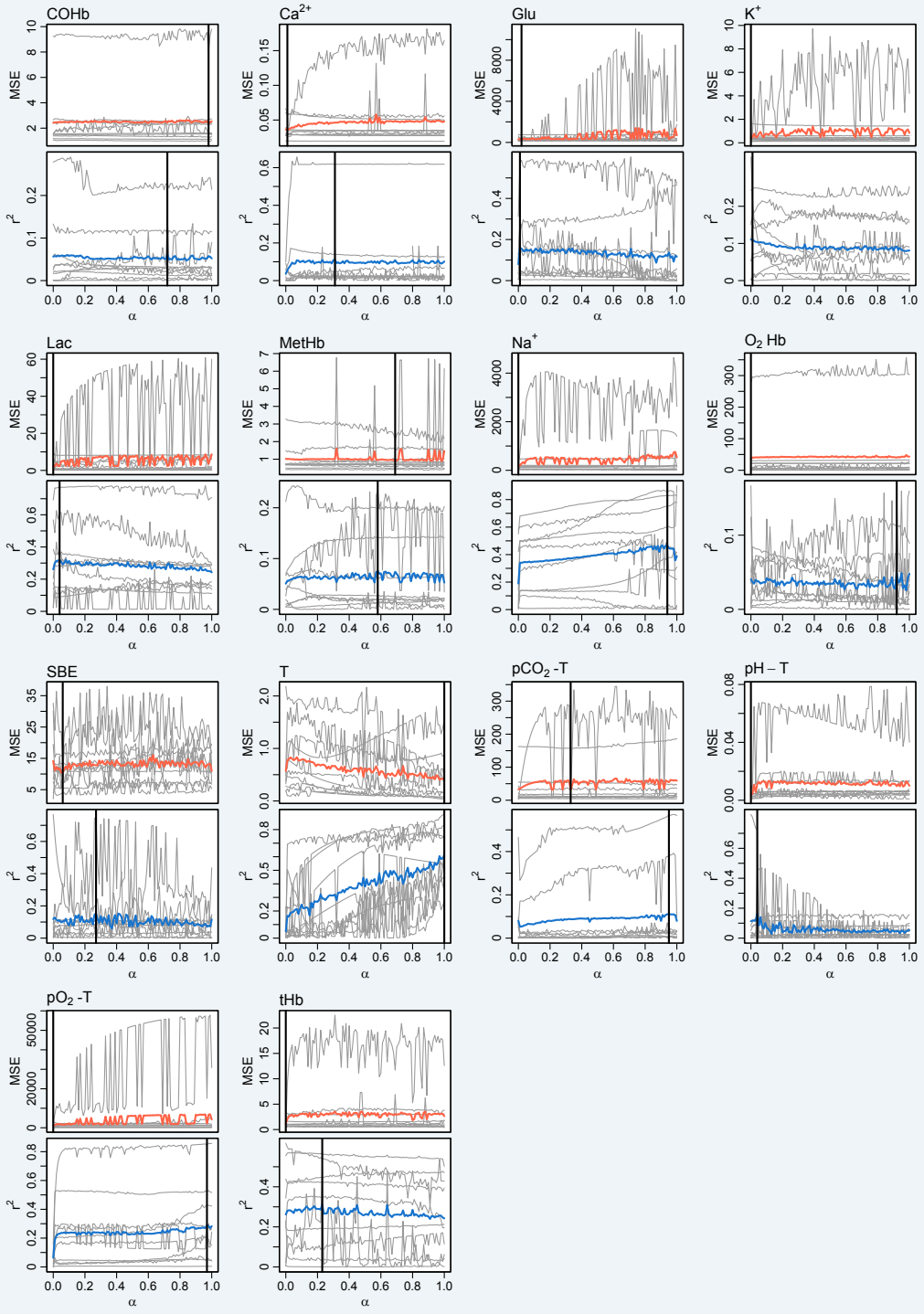


Figure 15.8: Evaluation of the optimal α values for reduction of MSE (top) and r^2 (below) for each of the 14 BGA parameters. The results of the individual crossfolds for animal subjects vna001–vna010 are shown in thin gray lines for each regression model, the average performance is shown in thick red and blue lines for MSE and r^2, respectively. The optimal alpha is highlighted by a vertical black bar.

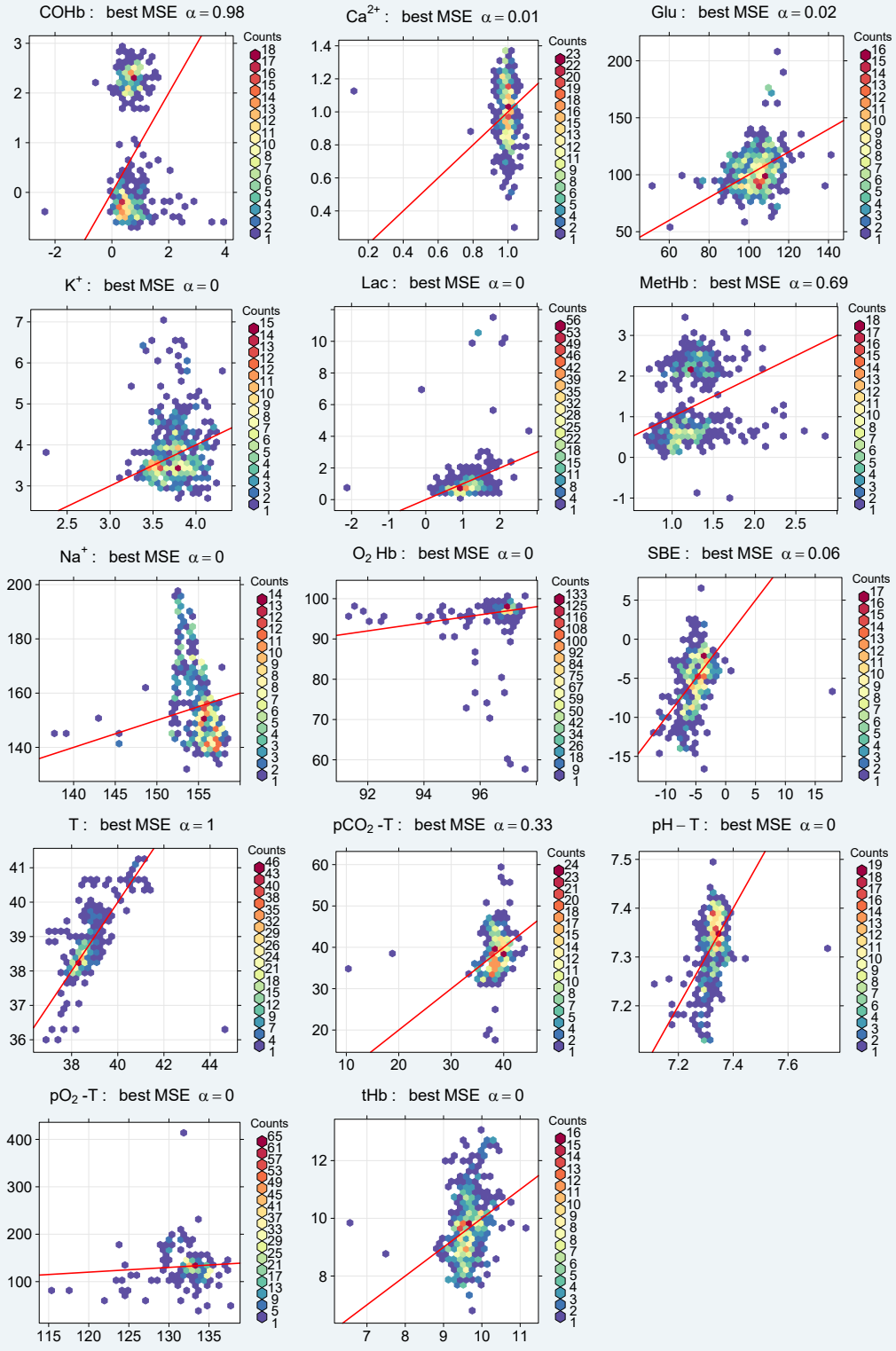


Figure 15.9: True (x-axis) vs. predicted (y-axis) values, based on the best α with minimal MSE for each parameter as hexagon binned density plots with a red reference line. Optimal predictions would be along this line. For COHb and MetHb, two separate clusters indicating distinct observed states can be observed.

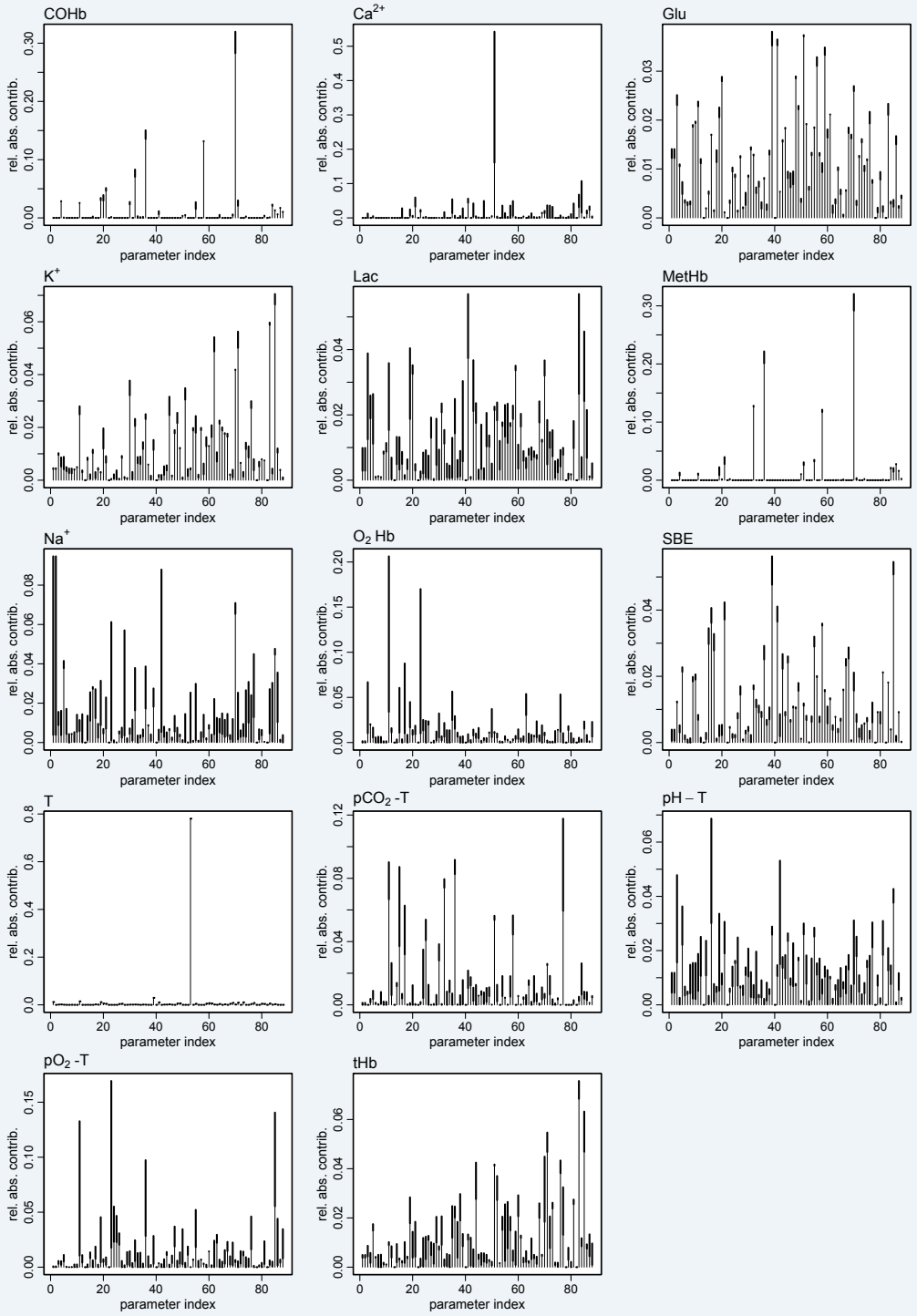


Figure 15.10: Visualization of the relative contributions of the regression coefficients for the evaluated parameters of animal subjects vna001–vna010. Differences between the mean MSE and mean r^2 model are shown as thick line segments for each input parameter.

Table 15.5: Influence of the surrogate parameters for a subset of blood gas parameters based on the relative contribution of the regression coefficients for *MSE* and R^2 optimization that are contributing to half of the weight of correlation coefficients.

Parameter	rel. influence MSE opt.	rel. influence r^2 opt.	Predictive Parameter
T	0.781	0.781	Philips MP50 IntelliVue blood temperature
	0.219	*0.219*	*remaining parameters*
FMetHb	0.320	0.292	Pulsion PulsioFlex O2er
	0.222	0.201	Osypka ICON str
	–	0.126	Osypka ICON pep
	0.458	*0.381*	*remaining parameters*
FCOHb	0.320	0.283	Pulsion PulsioFlex O2er
	0.150	0.136	Osypka ICON str
	0.132	0.131	Philips MP50 IntelliVue EVLW
	0.398	*0.450*	*remaining parameters*
cCa^{2+}	-0.161	-0.542	Philips MP50 IntelliVue Perf
	0.068	–	Jonosteril infusion rate
	0.054	–	Osypka ICON sqi
	0.049	–	Philips MP50 IntelliVue ABP dia
	-0.044	–	Osypka ICON vic
	0.042	–	Philips MP50 IntelliVue CFI
	0.042	–	Pulsion PulsioFlex dPmx
	0.036	–	Pulsion PulsioFlex PPV
	-0.036	–	Philips MP50 IntelliVue ITBV
	0.467	*0.458*	*remaining parameters*
pCO_2T	-0.083	-0.092	Osypka ICON str
	-0.074	-0.079	Osypka ICON pep
	0.067	0.090	Dräger Evita XL Inspiratory O2
	-0.059	-0.118	Pulsion PulsioFlex ScvO2
	-0.057	–	Philips MP50 IntelliVue EVLW
	-0.056	–	Philips MP50 IntelliVue Perf
	0.041	–	Osypka ICON ci-wt
	0.038	–	Osypka ICON icon
	-0.037	-0.087	Dräger Evita XL minimal airway pressure
	–	-0.063	Dräger Evita XL PEEP airway pressure
	0.488	*0.471*	*remaining parameters*
SBE	0.051	0.055	G20 infusion rate
	-0.048	-0.056	Osypka ICON tfc
	0.037	0.042	Dräger Evita XL respiratory rate
	0.036	0.041	Osypka ICON vic
	0.036	0.041	Dräger Evita XL minute-volume
	-0.036	-0.035	Philips MP50 IntelliVue EVLW
	-0.030	-0.035	Dräger Evita XL minimal airway pressure
	-0.029	-0.032	Philips MP50 IntelliVue SVV
	-0.028	-0.033	Dräger Evita XL PEEP airway pressure
	-0.025	-0.029	Pulsion PulsioFlex HP-PR
	-0.025	-0.029	Osypka ICON str
	-0.024	-0.026	Philips MP50 IntelliVue dPmax
	-0.023	-0.025	Pulsion PulsioFlex DO2I
	-0.022	-0.027	Philips MP50 IntelliVue CFI
	0.549	*0.495*	*remaining parameters*

15.2.4 Discussion

Parameter estimation and prediction are important aspects for closed-loop control of parameters dependent on blood gas analysis. As those measurements are only performed infrequently, often with several hours in between the measurements, the dynamic behavior of automated approaches is very restricted, limiting the achievable performance in homeostasis management. A sufficient prediction model utilizing surrogate parameters with higher temporal resolution may allow to bridge this gap by tighter control and faster feedback.

The first preliminary step for such predictions was the correct dating of the blood sample as variable withdrawal and analysis delays cannot be avoided and were present in the obtained measurements. As therapeutic interventions induced rapid changes in the vital parameters during the VNA study, no fixed window, e.g., observing and averaging the last 10 minutes prior to measurements, could be used but a correct identification of withdrawal times had to be performed with the developed detection algorithm to obtain a matched dataset. Containing all measurements of the VNA study, the dataset was preprocessed to remove unreliable parameters and incomplete measurement vectors.

To avoid an optimization bias toward individual parameters with large error potential or larger numerical ranges, scaling was needed as for ridge and lasso regression each error term is accounted equally. Looking forward to application within a real-time prediction system, no retrospective normalization may be used. Furthermore, training data may still contain significant outliers that may skew the scaling due to large differences in the numerical range. By using the 90 percent quantile as a scaling factor, those outliers were considered and a scaling to a nominal parameter range was achieved. This provided a more balanced training input for regression. Given this scaled data set, regression analysis was used to obtain estimates of the BGA parameters and evaluate the potential for parameter reduction. Missing parameters are an important problem for any prediction as only complete input vectors may be processed by a model, trained on those particular parameters. Thus, besides allowing for better clinical interpretation of the regression coefficients, a limited set of input parameters reduces the dependencies of regression models and allow for a more general application. Additionally, specific pre-processing steps may be used for the most significant and influential surrogate parameters regarding to the relative influence of their regression coefficients. With this goal in mind, methods like lasso and elastic-net regressions are a purposeful approach as the used L1 norm allows for these restrictions, whereas ridge cannot achieve this by using the L2 norm.

Evaluation of the predictive performance and the potential for parameter reduction was performed for the dataset obtained from the VNA study using elastic-net regression, including the edge cases of ridge and lasso regression. The effect of parameter reduction and potential relevant parameters to be pre-processed further was evaluated by variation of α as a weighted combination between lasso and ridge regression. The calculated β_{50} provided a good estimation of this parameter restriction potential, as the number of regression coefficients significantly contributing to the total regression vector can clearly be observed.

Yet, except for the temperature reading of the BGA, where a parameter reduction to a single input parameter yielded the best performance, parameter reduction only showed significant benefits regarding the predictive performance for $FMetHb$ and pCO_2T. For all other eval-

uated parameters, maintaining non-zero regression coefficients for all parameters yielded better results. This indicates that BGA parameters are dependent on a broader variety of influences or hidden, not observed factors, correlating to several observed parameters.
Another important assessment for clinical applications may be the performance regarding correlation and r^2 as often not the absolute numerical values are most relevant but relative changes and trends. Therefore, a model with good correlation would be more beneficial for this task than a model with a minimal MSE. To assess the influence of the MSE optimization regarding this goal, a comparison to the trained model with the best R^2 was performed. For most parameters, model selection yielded no significant differences for best α. Notable exceptions are the regressions for Na^+, Ca^{2+}, O_2Hb, and $pCO_2 - T$. Yet, for those parameters, achieved performance is almost equal for all values of α, thus selection was mostly based on small fluctuations of the achieved performance and not due to significant benefits of parameter reduction. This, limited by the achieved prediction accuracy, indicates that models of high quality with regards to the MSE also exhibit good correlation and thus a common model for both goals may be used.
Finally, a comparison to SVR with linear and Gaussian kernels was performed. The evaluations did not provide significantly different results compared to the elastic-net regressions and came with the disadvantage of interpretability. Whereas for linear regression methods like ridge, lasso, and elastic-net, the contributions of each parameter can be observed by the regression coefficients, such a direct evaluation of the contributing factors is not possible for SVR. Especially in the medical sector, where reason and causal relations play an important role for certification and approval, interpretability of the regression coefficients is an important criterion. Additionally, the SVR approach does not allow for parameter reduction, yet may be able to model non-linearity using Gaussian or polynomial kernels. Such more complex and non-linear methods may be required to train better models regarding surrogate parameters of blood gases. Yet, a significant increased performance would be required to justify this step, which could not be observed in the performed evaluation.
Whereas the performed analysis provided several insights regarding the required processing steps and achievable performance, some limiting factors and observations need to be discussed. Considering the results and the density distribution of the training data, a bias towards relatively constant observations is present. This effect becomes evident in the presented hexplots, where small, local clusters with a high density of similar measurements can be observed. Furthermore, within the hexplots of $COHb$ and $MetHb$, a bifurcation in two clusters of observations for individual animal subjects is present. These effects may be caused by the defined study conditions and the already successfully implemented closed-loop management. But presumably also by the body's own buffering abilities, stabilizing the current condition using biochemical receptors, cascades, and closed loops in order to maintain homeostasis. Thus, changes may only become visible if the body's buffering capabilities are exhausted. In addition, given only close to 500 training points, training capabilities were limited, especially for machine learning approaches like ANNs and SVR. To facilitate application of such methods, a much larger database with a greater number of measurements, including critical conditions and deviations from the stable state, would be required. This problem and the lack of data was probably aggravated by the required pre-processing to create a training dataset by removal of unreliable parameters and incom-

plete observations. As such observations may occur during critical clinical conditions, those edge cases may have been removed, facilitating a bias to stable measurements. Further processing by application of median filters to take account of measurement disruptions could possibly have also led to an impairment of the observed correlations.

15.2.4.1 Interpretation of regression coefficients

Given the optimal prediction results for the elastic-net regression performed for each parameter, the average weight of each regression coefficient can be used for interpretation of the performed regression. In what follows, this interpretation is performed for a few selected parameters where a significant contribution of individual regression coefficients was observed. For those parameters, the potential causal biochemical and medical relations between the regression coefficients and the parameters should briefly be discussed. If parameters with significantly contributing regression coefficients have such plausible causal relationships, this is a good sign for the feasibility of the model. However, if no such direct interpretation can be made, this does not invalidate the model as the observed parameters may still be influenced or correlated to the desired target parameter. Therefore, discarding a prediction model just because no clear interpretation can be made is unjustified [215].

15.2.4.1.1 Blood temperature T The arterial blood temperature was continuously measured by the patient monitor using a temperature sensor included in the PiCCO catheter system. This temperature was used for temperature calibration of the performed BGA by observation of the current temperature and manual input to the blood gas analyzer. Therefore, the blood temperature T provides a positive reference for the prediction method. Due to the manual input, however, only a limited accuracy and correlation is achieved since the temperature is roughly rounded to a maximum of one digit and therefore small fluctuations are not taken into account. When evaluating the regression coefficients of the optimal prediction, a dependence of about 78% can be observed for the individual measured temperature parameter. This is the expected correlation and the remaining 22% may be attributed to inaccuracies caused by manual processing.

15.2.4.1.2 FCOHb, FMetHb, FO_2Hb, ctHb A set of parameters obtained by the BGA are related to the hemoglobin (Hb) within the blood and used as indicators for different clinical conditions related metabolism and gas exchange. Carboxyhemoglobin ($FCOHb$) is a measure for CO bound to the red blood cells, Methemoglobin ($FMetHb$) is hemoglobin consisting of Fe^{3+} instead of Fe^{2+}, thus unable to bind oxygen. Oxyhemoglobin (FO_2Hb) is the relative fraction with bound oxygen. Furthermore, $ctHb$ is the total hemoglobin concentration, thus a measure for oxygen transport capacity. Regression analysis and parameter reduction yielded mostly similar regression coefficients related to a set of input parameters. Those parameters are related to gas exchange, ventilation, and cardiac parameters. The most significant regression coefficients related to hemoglobin were Pulsion PulsioFlex oxygen saturation (O2er), Osypka ICON's systolic time ratio (str), and the correlated preejection period (PEP) of the ECG. For the hemoglobin concentration $ctHb$, regression coefficients suggest negative correlation to infusion volume. As $ctHb$ is the relative amount of hemoglobin within the blood sample, dilution of the blood by supplementary infusions provides a causal relation.

15.2.4.1.3 cCa^{2+} For the calcium levels in the BGA results, the most significant regression factors were the perfusion index of the oxygen saturation monitoring (inversely correlated) and the Jonosteril infusion rate. Further significantly contributing factors include different parameters for cardiac function like ABP, variation of contractility index (vic), cardiac function index (CFI), and contractility index (dPmax). As Jonosteril contains 1.65 mmol/L Ca^{2+} [40], a causal relation to the infusion volume is obvious. Contributions of cardiac regression factors seems plausible as relations of measured calcium levels to cardiac work [216], ATP hydrolysis [217], or arterial pressure [218] have been reported. Regarding the negative correlation to blood flow of peripheral vessels, improved tissue perfusion has been correlated to inhibition of extracellular Ca^{2+} influx [219].

15.2.4.1.4 pCO_2T The temperature-corrected partial CO_2 obtained by BGA, is dependent on the regression factors of systolic time ratio (str), central venous oxygen saturation ($ScvO_2$), index of contractility (ICON), weight corrected cardiac index (ci-wt), and respirator settings. As respiratory parameters and cardiac work are related to CO_2, a causal relation of those regression coefficients to the measurement results is feasible.

15.2.4.1.5 SBE Regression coefficients of the optimal model for base-excess (SBE) are dependent on the respirator settings (RR, minute volume, airway pressures) and cardiac parameters. This suggests a relation to cardiac work and CO_2 production, influencing the base excess. Additionally, the thoracic fluid content (TFC) has a significant contribution to the regression. As this is a measure for the fluid state, a plausible link between the buffering capacity of the body and dilution related to a varying fluid content influencing the SBE is present.

15.2.4.2 Conclusion

The presented evaluation is only a brief example and first insight to the possibilities for prediction of blood gases using surrogate parameters, showing the benefit of a homogenous data collection and processing framework with high temporal resolution of a large number of vital parameters and settings.

Application of the presented regression pipeline with cross-fold validation for the task of predicting blood gases and for evaluation of parameter reduction could successfully be performed. Whereas the results only achieved a moderate goodness of fit regarding the evaluation of MSE and r^2, the positive reference for the observed blood temperature T could successfully be detected. In this case, lasso regression ($\alpha = 1$) provided the best result, essentially reducing the parameters to the single temperature input used for manual temperature correction of the BGA. Yet, only an r^2 of 0.61 could be achieved. This may be caused by inaccuracies in manual processing or time offsets and rounding the observed parameter prior to entering it manually at the blood gas analyzer.

For other parameters, only moderate performance regarding MSE and R^2 could be achieved. For most parameters, elastic-net regressions with different α weights had only a moderate effect on the performance and maintaining non-zero regression coefficients was beneficial. As no limited set of input parameters with non-zero regression coefficients could be selected, this indicates a complex dependency on the observed parameters or on unobserved ones, correlating to the observations.

Yet, for evaluated parameters where a restriction of the number of input parameters was beneficial, causal relations of the input parameters to most contributing regression factors can be made. Whereas the abundance of such interpretability cannot be used as an exclusion criterion for the regression model, causal relations strengthen the general feasibility of the approach for finding models based on surrogate parameters that can be used for closed-loop application.
The comparison of the best R^2 models to MSE models showed no significant divergence, requiring different models for predictions regarding the relative change and the absolute value. Yet, as the influence of α was small for most parameters and only moderate performance was achieved, this should be re-evaluated with better models to draw a conclusion if a general model for surrogate parameters covering relative changes and absolute numerical predictions may be used.

15.2.4.3 Outlook

Automated matching of BGA parameters with surrogate parameters may allow for a validation of each blood sample to deal with the problem of sample mix-ups [211, 212] and to be used as an estimator for physiological changes that need to be recognized and controlled more tightly to avoid life-threatening conditions. Surrogate parameters may be used for development of a dynamic BGA scheduler, providing the caregiver with a suggested timeframe for the next BGA depending on the current state and stability of the observed parameters. If for example, blood gas parameters are predicted to be changing, the suggested time for the next BGA could be reduced if those changes are of a significant degree. Caregivers may be notified to perform a BGA immediately to verify the predictions and provide adequate therapeutic measures to counter possible changes.

To achieve better performances in subsequent studies, effects within the time-domain and individual delays of the observed parameters might be considered as different parameters may have delayed influences on the blood gas parameters. Thus, not only the observations at a fixed point in time, related to the blood withdrawal, but a defined range of observations prior to blood withdrawal may be used. Additionally, the delays may be defined individually for each parameter to account for different physiological relations and delays.
Another limiting factor was the restricted number of usable data points with less than 500 measurements. Whereas selection of training data was performed to obtain the largest possible data set, individual parameters and measurements were excluded. This number of limited measurements and parameters paired with often only small measurement variations, e.g., for parameters like partial oxygen saturation, were limiting factors for the regression model. Thus, using a more extended data set with various different study conditions may lead to better results. Of course, limitation of the measurement observation variation is facilitated due to the already used closed-loops for homeostasis management.
Regarding the used regression method, novel and more robust versions of elastic-net regression (e.g., using Tukey's biweight criterion [220]) may be used to evaluate if a better and more reliable prediction based on the raw data stream from the medical devices can be achieved. Alternatively, more complex, non-linear models may be used. Whereas comparison to linear SVR models yielded similar results and short supplementary tests

with polynomial kernels could not lead to better results, different machine learning approaches like artificial neural network (ANN)s may be able to achieve this. Yet, such approaches need a significantly larger data base. This however, closes the loop regarding the initially stated problem of insufficient availability of medical data and often comes with the additional problem of being a *black box* solution, not allowing the assessment of causal relationships for the trained model. On contrary, by using regression-based analysis, the coefficients can yield a subset of influential parameters. Further research may then be focused on better and more robust filtering and sensor fusion to obtain more robust observations for those most relevant parameters.

Unfortunately, whereas more advanced methods may be able to reveal additional information contained within the observed data, or sensor fusion can achieve more reliable results, no method may be able to generate a surrogate model if the observed parameters may not contain enough information of the parameter to be estimated. This would require the desired parameter to be measured directly with novel and additional sensors. But even then, given the body's own closed-loop systems with most often excellent buffering capabilities, significant changes may still not be observable until a critical threshold, exceeding those buffering capabilities, is reached.

Part VI

Summary and concluding discussion

16 Summary of the thesis

In Chapter 1 a first introduction was given, highlighting the current issues regarding interconnection of medical devices with various protocols and the opportunities of automation in the context of clinical understaffing and an aging population.

In Part I with Chapter 2, the backgrounds for the relevant aspects of medicine and computer science were provided. For the medical background, this included the various medical devices, different aspects of patient management, the used medications, and the relevant aspects for animal studies as a surrogate model. Subsequently, the IT background provided a brief overview and introduction to the relevant topics of software design and networking. Finally, a brief introduction to fuzzy logic controllers was given and the relevant statistical and analytical aspects used for evaluation and regression analysis were presented.

In Chapter 3, the current state of the art with related work was presented. This included various examples for partial automation of individual devices, automated therapy management, and general concepts for medical device interconnection frameworks.

After those general introductions, the following parts of the thesis covered the performed studies, implementations, analyses, and results:
In Part II (Chapter 4) an overview of the conducted clinical studies with a porcine animal model was given to provide a general frame and reference for the implementation and results of the subsequent chapters.

In Part III the theoretical concept, the implementation, and the evaluation of the proposed framework for interconnection of medical devices and data collection in an experimental setting was presented. This framework uses a client-server architecture with various plugins for expanding the basic functions and a message-based approach for information transport. It was successfully evaluated during the two performed studies and able to collect over 1300 hours of scientific research data in a central database.

In Part IV, various applications based on the developed framework for management of hemo- and homeostasis to reduce variance in the study conditions and workload were presented:
In Chapter 11, the development and evaluation of automated anti-coagulation management was shown. It uses a modified measurement device and automated processing of the results to adjust a heparin infusion pump. This allowed for successful and automated management of the required heparin levels based on a state-of-the art look-up table for adaption during the performed studies.

In Chapter 12 another application, automated temperature management, was presented and evaluated. Using a modified forced-air warming device to allow for remote-control by communication with the developed framework, TTM based on the body temperature was implemented. Evaluation showed that the desired nominal body temperature could be maintained at any time, drawing level with the performance achievable by constant human supervision and frequent manual adaption to reduce thermal effects on homeostasis.
In Chapter 13, the proposed adaptive and multi-purpose fuzzy-logic controller was presented. It uses a relative output factor for adaption, allowing a dynamic response and tolerating human interference. Evaluation of the controller was performed during the studies by automating the management of $etCO_2$, glucose, and electrolytes to improve homeostasis. For the frequently adaptable RR for management of $etCO_2$, target levels could be maintained in over 95% of the total observation time. Glucose could be kept in a nominal range in over 85% of the time. For the electrolytes, evaluation was limited to individual animal subjects with an average performance.

In Part V, retrospective analysis based on the collected data with a focus on the problem of infrequently obtainable BGA results was shown:
In Chapter 14, a brief overview of the data collection and the general approach to gain a first insight into the collected data was presented. This was especially relevant, as it allowed for pre-processing and detecting patterns and characteristics relevant in further processing steps.
In Chapter 15, one example for such a characteristic pattern within the obtained data was shown and used for the automated detection of blood withdrawal events to allow for an automated and improved sample dating. For the 886 BGAs performed during the studies, a F1-score of 0.97 could be achieved. Those detected events could then be used to solve the problem of matching the obtained blood samples to other vital parameters. Given the observed problem of closed-loop management with sparse adaption intervals due to the limited number of blood samples from BGA, a regression analysis to assess the potential of surrogate parameters for blood parameters, utilizing the calculated withdrawal times, was performed. This initial analysis showed a limited performance on the available dataset, suggesting a restricted predictability and the requirement for an increased and more comprehended training data set with more diverse physiological conditions.

17 Concluding discussion

Research in the scope of clinical decision making and automation is a constantly evolving field with a steadily increasing number of publications in recent years. Whereas progress in clinical techniques and equipment allow for treatment of more complex and life-threatening conditions [1], technological advantages and automation are still mostly limited to patient management and guidance systems [3]. For potential improvements in automation technology, better interconnection of medical devices is an essential step towards solving the fundamental problem of clinical decision making: the availability of data.
Since clinical decision support systems often have none or limited access to the displayed data, information cannot be collected and processed [179]. Thus, possible knowledge from clinical practice is still lost forever on a daily basis [9] and most knowledge and insights for decision-making need to rely on empirical research [2]. When solutions for clinical information systems and automation are proposed, they are often limited to theoretical concepts and the evaluation is limited to the few publicly accessible collections of clinical data, often only containing short time frames and a limited number of parameters, or performed entirely on simulated data.
For bridging this data gap, a unified system for interconnection and automation of medical devices, which was previously only a theoretical concept, has now been evaluated in the research environment for porcine studies at the UKT and was used to collect significant amounts of research data. During the studies presented in the context of this work, fundamental solutions and concepts for an ICU software framework for medical devices were developed and evaluated as the initial and most important step, providing the main focus of this thesis.
Automation and closed-loop systems, which were defined and guided by the clinical questions and practical requirements in the used experimental porcine ICU, were implemented and have been evaluated successfully in this real setting based on the developed framework. A central concept for the implemented automation was the usage of a fuzzy logic-based controller. Fuzzy logic was selected as the method of choice, as it is a well-established concept and empirical clinical decision making is often based on linguistic descriptions that are intuitive to physicians. Such descriptions include the often fuzzy range of commonly acceptable physiological values, which need to be assessed individually to obtain optimal settings [3].
Whereas other approaches utilizing fuzzy logic for clinical decision making use absolute relations for the membership functions, the developed fuzzy controller is based on relative factors applied to the current setting of various medical devices. Since the factors used are relative, automated therapy can be started and adjusted at any time and in any condition. In addition, this used approach is able to tolerate human interactions and reconfigurations, which are an important aspect during therapy, repeatedly observed in the carried-out studies.

The feasibility of this adaption method and the system in general could be demonstrated with the successful implementation of adaptive fuzzy-logic controllers for management of expiratory CO_2, glucose, and several electrolytes. They provide an initial proof of concept and may form the basis for further improvements and studies in this ICU setting with more adaptive algorithms, considering further interactions.
Whereas many specific solutions for medical devices or features exist, such research and development is mainly focused on and limited to individual components and aspects, specifically tailored for individual devices or manufacturers. Therefore, no such general concept is required and application is most often limited to individual laptops or the scope of the individual medical device. Furthermore, most available systems can only be used for collection of clinical data in a unidirectional way. Of course, given that such systems are in general medical products, this necessitates legal and functional limitations of scope and performance for their respective tasks and mostly prohibits direct control capabilities.
For automation and control, however, it is not sufficient to simply collect data, but systems that provide a return channel for the general control of connected devices are required. Aside from specific implementations for individual tasks, only some approaches for general medical frameworks have been postulated over the recent years. Yet, published literature is mostly limited to theoretical concepts or assessments of the emerging demand for such systems. Additionally, none of the proposed system is specifically tailored to the needs of academic research.
During development of the software framework, the closed-loop systems and the practical implementation of such a fully-functional system in a research environment as a bridge between theoretical concepts and clinical application, various difficulties that have not been covered by the various theoretical concepts became noticeable. Maybe the most important of those aspects was the physical implementation regarding required hardware, network structure, and storage that is required in an experimental or clinical setting.
Whereas clinical application of such a system and the developed solutions are ultimately the long-term goal, the necessary legal framework restricts the development and evaluation of such systems. For clinical research in animal studies, such legal limitations are not as strict as those for human participants, thus allowing for a more comprehensive evaluation under human supervision. As this was the available experimental context for this research, a novel, easily expandable experimental framework, aimed at overcoming those limitations and providing functional implementation for evaluation and further academic research, could be developed. It is specifically tailored for medical applications with its very heterogeneous landscape of devices and able to collect a broad variety of data from different sources. The message-based system allows for arbitrary bidirectional communication between connected devices. This facilitates easy expansion and evaluation of novel sensors, methods, closed-loop systems, and collecting data on various interactions in changing study conditions. Which might provide insights and help developing concepts for the future of critical care as already shown successfully for various automated tasks during the performed studies. Yet, besides evaluation of algorithmic solutions, the physical implementation of such a system was an important step to investigate interactions with caregivers and how such a system can be integrated into the medical work flows as an assistive technology.

As the body is a very complex manifold system with highly sophisticated interactions, this need is further enhanced as simple therapeutic protocols can hardly be developed and only provide a limited quality of care [3]. Especially in critical conditions, optimal levels cannot be defined in general but must be assessed individually with as much information as possible [3]. Moreover, patient care is highly dynamic and clinical decision making is time sensitive [179]. Automated systems may rapidly interact with the patient and be able to better assess complex situations and recognize changes in patient condition earlier, allowing for faster and more proactive interventions [3, 179]. Additionally, such systems can reduce therapeutic variances between patients, if two or more medically acceptable solutions exist for a single condition [2]. This may be especially beneficial, if no continuous therapy would have been provided by switching the goals in an ongoing therapy, e.g., due to shifting medical staff.
Independently of the individual goals regarding improved care, personalized medicine, smarter alarms, and automation using machine learning, the fundamental requirement for clinical research leading to better, personalized medicine is a large and sufficient high-quality data foundation. Unfortunately, such data is rarely available and most often limited to short periods or few parameters. Using the developed framework and database, this kind of data could be collected in the used experimental setting. It provided a research foundation containing over 1000 hours of observations for 300 synchronized parameters in a single unified database format, exceeding any other available data source.

Yet, even when fundamental problems for interconnection and data processing are solved as it is the case for the developed solutions presented within this thesis, still no continuous data stream for several parameters can easily be established. An example for this scenario are measurements of blood gases, which could only be obtained with a limited frequency. For the fuzzy logic-based glucose controller, short intervals are needed and the limitation provides a general problem in glucose management [147]. With this dependency on infrequent and manually performed measurements, there is an imminent need for finding alternative methods to estimate such parameters and interpolate the time between measurements, allowing to detect critical situations. Additional challenges regarding the availability and quality of data can be demonstrated by using this example as well. Not only is it difficult to collect information from various medical devices to be used as surrogate parameters and matching them in the time domain to get a complete measurement vector of the current state, but finding the correct time point for prediction provides an additional challenge. As there is no specific parameter for blood withdrawals events that can easily be established without the need for manual interaction, it had to be derived automatically from the given parameters as shown with the proposed detection for withdrawal events. Only then further analysis like the evaluation of surrogate parameters with a correctly matched input vector can be performed.
However, even with the usage of surrogate parameters and predictive models, automated systems are dependent on sufficient data input. If a sensor fails, controls can be restricted or become impossible. An example is the respiratory management during the conducted studies, where the developed fuzzy closed-loop controller is solely dependent on the

ventilators $etCO_2$ sensor input. During the observation of two animal subjects, problems with the sensor caused disruptions and measurement errors, forcing the automated system to be stopped to prevent possible health risks. Thus, this example of a relatively simple closed-loop system already illustrates the importance of sensor fusion with redundant parameters in control algorithms.

Beside improvement of future critical care with personalized medicine, another important aspect and motivation for automation is the rising demand for critical care in an aging population with a limited number of caregivers and thus, increased workload [3, 4]. By using advanced technologies with an increasing number of devices, observed parameters, and POC tests, clinical decision making can be improved. However, such tools and their protocols may become more and more complicated and difficult instead of being part of clinical routine [1]. This complexity will increase workload, lead to more errors, and reduced compliance [3]. Automation may have a significant impact in dampening this effect as there are still many repetitive and manual tasks where workload and complexity could be improved to boost compliance and significantly reduce the risk of human errors [212]. An examples for such a task is automated processing of BGA measurements for anti-coagulation management, using the ACT device presented within this thesis. Automating the protocol-based manual approach for adaption of the heparin pump with a closed-loop application embedded in the developed framework, reduction of workload and error potential during the calculations could be achieved.
Another concern regarding the increasing number of different devices and monitors is the risk of missing crucial information as human observers might get overwhelmed by the amount and variation of the presented information. Whereas IT and automated systems in principle should ease this burden, compartmentalization and limitations of the individual devices lead to increased workload [3]. As this is a problem concerning the medical environment as a whole, automation of individual aspects of care within their respective devices and monitors cannot lead to a reduction of workload as parameters, and now additional settings, still need to be observed and controlled at their associated interfaces.
Therefore, general concepts including data aggregation and presentation are required. Only systems that can interact with many devices and collect data from different sources to gain a detailed insight into the patient's condition and allow adaptations can achieve this goal. Such central interfaces, where all information about the patient is gathered and therapeutic measures can be carried out, will become even more important as the amount of data continues to grow. For the presented experimental research system, such a central observation and control monitor was already an essential aspect of the treatment. By using a touch-screen with a freely implementable GUI, interaction with the system and the different interface types for monitoring and control of medical devices, therapeutic protocols, and closed-loop systems by caregivers was possible and could be evaluated.
One advantage of automated systems is not requiring continuous observation once a therapeutic goal has been set. As in many settings and scenarios supervision by caregivers cannot be assured at all times, this advantage may be particularly beneficial as various therapies may be extended to settings where the required observation is not currently available. An example would be non-invasive closed-loop temperature management within

ICUs, where unsupervised application may cause patient harm. Evaluation of such a task was performed for closed-loop temperature management using a forced-air warming system. By adding a monitoring and control interface and developing a closed-loop system for automated temperature management, the workload for temperature management has been significantly reduced and even better performance than with manual management has been achieved to improve homeostasis maintenance.

Yet, whereas providing many advantages and, above all, having the potential to reduce the amount of work, automation carries some risks. As automated systems perform therapeutic measures independently, physicians and caregivers may no longer take notice of developing and occurring physiological changes. This might limit evidence-based research and advances in medicine, as the ability and motivation to detect and understand pathophysiological changes could vanish [3]. Thus, possibly inappropriate therapies carried out by automated systems might go unnoticed for a prolonged period of time. Besides the need for maintaining interaction and awareness of the caregivers by providing automated systems as tools for an improved goal-directed and personalized care, this clearly shows the need for a concurrent development of smarter alarms to detect such systems and alert caregivers.
In general, automation should not be a competition to caregivers, but be developed and seen as a form of displacement, instead of considering it as a replacement, allowing caregivers to focusing on therapy guidance and other tasks where human experts are more flexible and can exceed the capabilities of algorithms. Above all, caregivers should not be forced to accept decisions that contradict their expertise and critical thinking [1], as this reduces acceptance and, if implemented in this way, could significantly limit interaction and awareness with regard to automated decision-making.

Besides all technical and psychological difficulties, there is still a third important obstacle as a limiting factor for automation in the medical field. Certification of medical devices is already a lengthy and expensive process. Considering decision-making in closed-loop and full-automated systems, many questions regarding the certification process and liability remain unanswered. In the near future, applications may still be limited to suggestion- and open-loop systems. However, manual processing or entering of data from different devices might finally be replaced by automated processing, just requiring human verification. However, these verification tasks can be error-prone or even become more problematic unless adequate observation and critical decision making is ensured, especially if it becomes routine to accept the proposed results of the automated system. Thus, human verification for each automated action in comparison to a straightforward implementation of a fully automated solution with sufficient auto-validation of the results may not always be preferable.
Given this uncertainty about changes in medical regulatory frameworks and liability issues, healthcare insurers who already cover the cost of treatment and, in the future, may even cover treatment errors by automated systems could play a key role. Watching the developments in other industrial sectors, such as the automotive one, with car insurers as an equivalent type of a public insurer dealing with questions of assisted or autonomous driving could play a guiding role for dealing with automation issues in the medical sector. Yet, the scenario of autonomous driving is only a conditionally fitting example as drivers

strive to focus on other tasks instead of monitoring the car, whereas in other areas with automation, for example pilots or also caregivers are employed to ensure safety and, if necessary, intervene in order to prevent and deal with critical situations. Moreover, if such automated systems are considered as assistive systems, physicians will continue to set therapeutic goals and be responsible for supervising the therapy.

17.1 Outlook

Given the developed framework and the evaluation of first closed-loop systems for automated intensive care medicine, further research may focus on various different aspects, enhancing the performance and scope of this first, fundamental implementation and evaluation. The development and implementation of physical systems for interconnection is an essential aspect and will become even more important with an increasing number of medical devices and therefore should be a research focus in this fundamental stage of clinical automation. Hence, requiring to elaborate and evaluate solutions regarding the scalability of medical frameworks and their application to multiple hospital wards. This will require considerations regarding redundancy, sufficient data storage, and all other practical aspects for the implementation in every day clinical practice. Otherwise, although there may be elaborated concepts for decision making within a legal framework, the practical implementation and management of those systems within the clinical environment will still remain an unsolved problem for a prolonged time.

Another important research aspect might be the reduction of caregivers' workload and further (semi-) automation of the various manual tasks in intensive care settings. The provided example for automated processing of ACT measurement results using an existing therapeutic protocol was just a proof of concept and a first step for further research towards more advanced therapeutic strategies and for other POC and laboratory devices. Likewise, the automated detection of blood withdrawals is not only an important first step for analysis and automation, but it can also facilitate easier patient selection by caregivers and the detection of sample mix-ups. This may include the better integration to BGA devices by providing reduced and sorted patient lists, dependent on the detected blood withdrawals, to improve patient safety and reduce workload at the same time.

Based on the data collected with the developed framework during the conducted studies, in-depth analysis regarding the clinical research questions of the studies, e.g., the analysis of patient's volume state based on the collected monitoring data, may be performed. Besides monitoring data collected once per second, real time curves with up to 500 Hz have been collected from patient monitor and ventilation device. This data may be utilized and allow for ECG and respiration pattern analysis using different approaches like wavelets and machine learning to segment the data stream and develop detection and classification algorithms for different physiological states. Applied to the detection of critical conditions and complications, this may lead to refinements of study protocols, significantly reducing the risk of errors and possible harm to the study animals. Combined with the developed automated solutions for stabilizing the animal subjects and standardizing the study conditions, this could lead to a reduction of the number of animal subjects required to obtain significant

study results. Obtained results may then lead to improvements of guidelines and therapies in clinical practice.
Furthermore, given the growing collection of high resolution data for various conditions, development of better models, predictions, and simulation may be performed. Additionally, using an extended database, research questions might be retrospectively answered to an increasing degree based on the collected observations. Even regarding parameters that were unrelated to the initially performed study but possible by collecting all available information and preserving it for such a potential future use. This might provide a useful tool as a replacement for studies to actually be carried out on animals. Obtained research results may also be directly applicable to humans and could lead to improved care during the current research phase, where the technical and legal frameworks still need to be established to allow more advanced automation in clinical settings.
Additionally, given the extensive set of collected parameters in the used experimental ICU setting, research might focus on the aspect of sensor fusion for parameters from different devices. Given an ensemble of medical devices, several parameters like HR and others are measured by different systems and methods. Thus, sensor fusion might be used to generate a more reliable parameter from the raw data stream for further processing and analysis. Within this context, the development and integration of additional sensors and information sources to facilitate sensor fusion might be relevant, too. This includes better sensing technology for different aspects like continuous glucose monitoring (CGM), which is still experimental or too expensive for a disposable product in clinical practice, and may provide alternatives to the indirect estimation of BGA parameters based on patient monitoring data.
However, even if such training data is available for development of better models and algorithms, its application as a tool for personalized medicine will still be mainly dependent on a continuous and comprehensive stream of monitoring information from the individual patient to provide optimal decision-making and aiding caregivers as the body itself is a regulatory closed-loop system that is able buffer and compensate numerous conditions.
Besides the already variable therapeutic settings for each patient, a trend for further compartmentalization into specific care units, with their respective set of devices, can be observed with the increasing number of specific systems and therapies. Even with the best possible device interconnection, such restrictions in clinical settings, providing only a subset of parameters, should be considered regarding the choice of models and the required input parameters as it will provide additional problems and require further research for automation and decision-support systems. Yet, such restrictions are even to be considered from a clinical standpoint, as they should be avoided because patients may develop the same critical conditions independently of the assigned special unit [3].
Another important aspect of automation and integrated systems in the medical field will be the need for better data presentation on a single screen, as the rising number of devices and monitors make keeping an overview more and more difficult. Additional research is required to find practical and intuitive ways for interaction with automated control systems, like the fuzzy controller, to allow caregivers to easily set goals and parameters for the desired therapy. The importance of this aspect became clear during implementation and evaluation of the rate of change as an additional fuzzy function for calculation of the adaption factor. Whereas the slope could of course have been used, it is not a common unit of measurement

in the clinical environment compared to time spans. Those are a more natural measure for physicians in clinical practice. The search for such measures of configurable parameters, which can be assessed by physicians through empirical knowledge or practical need, is therefore an important aspect that should be taken into account in every implementation and further developments. Furthermore, additional research needs to consider and evaluate human interactions, interface design in the context of large amounts of information. This will require smart filtering of information for context-aware presentation of the currently most relevant and important information to caregivers and physicians in order to reduce information density but still allow a fast assessment of the current patient state.

17.2 Conclusion

Given those various open questions, sparsity of data, and technical limitations, automated clinical care is still a long way off. The developed experimental framework and evaluations for different closed-loop and data collection solutions may provide essential steps for further research regarding automation and surrogate parameters. Yet, including consideration of the current legal limitations and potential risks, much additional research and a better legal foundation is required.
Maybe the most suitable area of application for automation and closed-loop strategies at this time are remote locations where telemedicine used, e.g., on offshore structures or other isolated locations where no qualified physicians are in reach [3, 221]. Once more, the most important aspect for this task is the availability of all collected data to the physician. By providing automated therapy where expertise or staffing is otherwise not sufficient, qualified physicians may be able to remotely guide an automatically performed goal-directed therapy. Yet, besides any long-term goals, possible clinical application areas, and remaining questions, the development and implementation of the proposed system provides an imminent improvement for further research regarding animal studies in the established experimental setting at the UKT and is the foundation for several medical doctoral theses and research papers. Additionally, as all available information is collected and information loss is minimized, retrospective analysis based on all observations can be performed.

Finally, it is time to close the loop and return to the introductory question — *Would you entrust your life to a computer while staying critically ill in a hospital?* — Given the current state of the art and the research presented within this thesis, this fundamental question is still mostly hypothetical as many aspects and difficulties regarding clinical application of automation and machine learning remain to be addressed. Yet, machine-learning and automation are already beginning to find applications in various medical devices. Considering the rising demand for an aging population and the progress in medical technology for better treatments and patient outcome, it will just be a matter of time until advanced systems with better interconnection will become part of everyday clinical practice. This will most likely be a continuous process with more and more assistive technologies. Thus, within the next decades, the hypothetical question may just turn into a rhetorical one, as everybody may get used to physicians and caregivers piloting the journey to recovery.

18 Publications arising from this thesis

- J. Peter, W. Klingert, A. Königsrainer, W. Rosenstiel, M. Bogdan, and M. Schenk. "TICoMS – A Modular and Message-Based Framework for Monitoring and Control of Medical Devices". In: *27th International Symposium on Computer-Based Medical Systems (CBMS), Proceedings of*. IEEE, 2014, pp. 473–474. ISBN: 978-1-4799-4435-4. DOI: 10.1109/CBMS.2014.96

- J. Peter, W. Klingert, M. Spüler, A. Königsrainer, W. Rosenstiel, and M. Schenk. "Automated therapeutic anticoagulation: A closed-loop approach using a modified measurement device". In: *Biomedical Engineering (BioMed), 13th IASTED International Conference on*. IEEE. 2017, pp. 224–228. ISBN: 978-0-88986-990-5. DOI: 10.2316/P.2017.852-027

- J. Peter, W. Klingert, K. Klingert, K. Thiel, D. Wulff, A. Königsrainer, W. Rosenstiel, and M. Schenk. "Algorithm-based arterial blood sampling recognition increasing safety in point-of-care diagnostics". In: *World Journal of Critical Care Medicine* 6.3 (2017), pp. 172–178. ISSN: 2220-3141. DOI: 10.5492/wjccm.v6.i3.172

- W. Klingert, J. Peter, C. Thiel, K. Thiel, W. Rosenstiel, K. Klingert, C. Grasshoff, A. Königsrainer, and M. Schenk. "Fully automated life support: An implementation and feasibility pilot study in healthy pigs". In: *Intensive Care Medicine Experimental* 6.2 (2018), pp. 1–12. ISSN: 2197-425X. DOI: 10.1186/s40635-018-0168-3

- J. Peter, K. Klingert, W. Klingert, A. Königsrainer, C. Grasshoff, W. Rosenstiel, and M. Schenk. "Automated closed-loop management of body temperature using forced-air blankets: preliminary feasibility study in a porcine model". In: *BMC Anesthesiology* 18.80 (2018), pp. 1–11. ISSN: 1471-2253. DOI: 10.1186/s12871-018-0542-4

Bibliography

[1] E. Brown, J. Clarke, K.-L. Edward, and J.-A. Giandinoto. “Point-of-care testing of activated clotting time in the ICU: is it relevant?” In: *British Journal of Nursing* 25.11 (2016). ISSN: 0966-0461. DOI: 10.12968/bjon.2016.25.11.608.

[2] J. E. Wennberg. “Unwarranted variations in healthcare delivery: implications for academic medical centres”. In: *British Medical Journal* 325.7370 (2002), p. 961. ISSN: 0959-535X. DOI: 10.1136/bmj.325.7370.961.

[3] J.-L. Vincent. “The future of critical care medicine: integration and personalization”. In: *Critical care medicine* 44.2 (2016), pp. 386–389. ISSN: 0090-3493. DOI: 10.1097/CCM.0000000000001530.

[4] D. C. Angus, M. A. Kelley, R. J. Schmitz, A. White, J. Popovich Jr, and Committee on Manpower for Pulmonary and Critical Care Societies (COMPACCS). “Current and projected workforce requirements for care of the critically ill and patients with pulmonary disease: can we meet the requirements of an aging population?” In: *JAMA* 284.21 (2000), pp. 2762–2770. ISSN: 0098-7484. DOI: 10.1001/jama.284.21.2762.

[5] M. McHugh, F. McCaffery, and V. Casey. “Software process improvement to assist medical device software development organisations to comply with the amendments to the medical device directive”. In: *IET software* 6.5 (2012), pp. 431–437. ISSN: 1751-8814. DOI: 10.1049/iet-sen.2011.0198.

[6] M. Zema, S. Rosati, V. Gioia, M. Knaflitz, and G. Balestra. “Developing medical device software in compliance with regulations”. In: *Engineering in Medicine and Biology Society (EMBC), 37th Annual International Conference of*. IEEE. 2015, pp. 1331–1334. ISBN: 978-1-4244-9271-8. DOI: 10.1109/EMBC.2015.7318614.

[7] A. E. Johnson, M. M. Ghassemi, S. Nemati, K. E. Niehaus, D. A. Clifton, and G. D. Clifford. “Machine learning and decision support in critical care”. In: *Proceedings of the IEEE* 104.2 (2016), pp. 444–466. ISSN: 0018-9219. DOI: 10.1109/JPROC.2015.2501978.

[8] P. G. Barash, B. F. Cullen, R. K. Stoelting, M. Cahalan, M. C. Stock, R. Ortega, and S. R. Sharar. *Clinical Anesthesia Fundamentals*. Wolters Kluwer Health, 2015. ISBN: 9781496310842.

[9] L. A. Celi, R. G. Mark, D. J. Stone, and R. A. Montgomery. “"Big data" in the intensive care unit. Closing the data loop”. In: *American journal of respiratory and critical care medicine* 187.11 (2013), pp. 1157–1160. ISSN: 1073-449X. DOI: 10.1164/rccm.201212-2311ED.

[10] K. Lesh, S. Weininger, J. M. Goldman, B. Wilson, and G. Himes. “Medical device interoperability – assessing the environment”. In: *Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability. (HCMDSS-MDPnP)*. IEEE. 2007, pp. 3–12. DOI: 10.1109/HCMDSS-MDPnP.2007.22.

[11] O. Badawi, T. Brennan, L. A. Celi, M. Feng, M. Ghassemi, A. Ippolito, A. Johnson, R. G. Mark, L. Mayaud, G. Moody, C. Moses, T. Naumann, V. Nikore, M. Pimentel, T. J. Pollard, M. Santos, D. J. Stone, A. Zimolzak, and MIT Critical Data Conference 2014 Organizing Committee. “Making big data useful for health care: a summary of the inaugural mit critical data conference”. In: *JMIR medical informatics* 2.2 (2014), e22. ISSN: 2291-9694. DOI: 10.2196/medinform.3447.

[12] O. Caelen, G. Bontempi, E. Coussaert, L. Barvais, and F. Clément. “Machine learning techniques to enable closed-loop control in anesthesia”. In: *Computer-Based Medical Systems (CBMS), 19th IEEE International Symposium on*. IEEE. 2006, pp. 696–701. ISBN: 978-0-7695-2517-4. DOI: 10.1109/CBMS.2006.110.

[13] J.-L. Vincent. “Critical care – where have we been and where are we going?” In: *Critical Care* 17.1 (2013), S2. ISSN: 1364-8535. DOI: 10.1186/cc11500.

[14] N. A. Halpern and S. M. Pastores. “Critical care medicine in the United States 2000–2005: an analysis of bed numbers, occupancy rates, payer mix, and costs”. In: *Critical Care Medicine* 38.1 (2010), pp. 65–71. ISSN: 0090-3493. DOI: 10.1097/CCM.0b013e3181b090d0.

[15] J. D. Bronzino. *The Biomedical Engineering Handbook*. 3rd ed. Electrical Engineering Handbook. Taylor & Francis, 2006. ISBN: 9780849321245.

[16] P. L. Marino, G. Geldner, and T. Müller-Wolff. *Das ICU-Buch: Praktische Intensivmedizin*. 5th ed. Urban & Fischer Verlag/Elsevier GmbH, 2017. ISBN: 978-3437231629.

[17] T. M. Enomoto and L. Harder. “Dynamic indices of preload”. In: *Critical Care Clinics* 26.2 (2010), pp. 307–321. ISSN: 0749-0704. DOI: 10.1016/j.ccc.2009.12.004.

[18] C. of the European Union. *REGULATION (EU) 2017 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EEC*. 2017. URL: http://data.consilium.europa.eu/doc/document/ST-10728-2016-INIT/en/pdf (visited on 2017-04-25).

[19] P. G. Hattersley. “Activated coagulation time of whole blood”. In: *JAMA* 196.5 (1966), pp. 436–440. ISSN: 0098-7484. DOI: 10.1001/jama.1966.03100180108036.

[20] E. L. Lewandrowski, E. M. Van Cott, K. Gregory, I.-K. Jang, and K. B. Lewandrowski. “Clinical Evaluation of the i-STAT Kaolin Activated Clotting Time

(ACT) Test in Different Clinical Settings in a Large Academic Urban Medical Center: Comparison With the Medtronic ACT Plus". In: *American Journal of Clinical Pathology* 135.5 (2011), pp. 741–748. ISSN: 0002-9173. DOI: 10.1309/AJCPSF8ASGONNQM6.

[21] D. Fitzgerald, A. Patel, S. Body, and S. Garvin. "The relationship between heparin level and activated clotting time in the adult cardiac surgery population". In: *Perfusion* 24.2 (2009), pp. 93–96. ISSN: 0267-6591. DOI: 10.1177/0267659109106729.

[22] M. E. Nunnally, R. Jaeschke, G. J. Bellingan, J. Lacroix, B. Mourvillier, G. M. Rodriguez-Vega, S. Rubertsson, T. Vassilakopoulos, C. Weinert, S. Zanotti-Cavazzoni, and T. G. Buchman. "Targeted temperature management in critical care: A report and recommendations from five professional societies". In: *Critical Care Medicine* 39.5 (2011), pp. 1113–1125. ISSN: 0090-3493. DOI: 10.1097/CCM.0b013e318206bab2.

[23] A. Torossian. "Survey on intraoperative temperature management in Europe". In: *European Journal of Anaesthesiology* 24.8 (2007), pp. 668–675. ISSN: 0265-0215. DOI: 10.1017/S0265021507000191.

[24] V. D. Hooper and J. O. Andrews. "Accuracy of noninvasive core temperature measurement in acutely ill adults: the state of the science". In: *Biological Research for Nursing* 8.1 (2006), pp. 24–34. ISSN: 1099-8004. DOI: 10.1177/1099800406289151.

[25] S. Barnason, J. Williams, J. Proehl, C. Brim, M. Crowley, S. Leviner, C. Lindauer, M. Naccarato, A. Storer, and A. Papa. "Emergency nursing resource: non-invasive temperature measurement in the emergency department". In: *Journal of Emergency Nursing* 38.6 (2012), pp. 523–530. ISSN: 0099-1767. DOI: 10.1016/j.jen.2012.05.012.

[26] D. L. Longo, A. S. Fauci, D. L. Kasper, S. L. Hauser, J. L. Jameson, and J. Loscalzo. *Harrison's Principles of Internal Medicine*. 18th ed. McGraw-Hill Education, 2011. ISBN: 9780071748902.

[27] D. I. Sessler. "Mild perioperative hypothermia". In: *New England Journal of Medicine* 336.24 (1997), pp. 1730–1737. ISSN: 0028-4793. DOI: 10.1056/NEJM199706123362407.

[28] D. I. Sessler. "Temperature monitoring and perioperative thermoregulation". In: *Anesthesiology* 109.2 (2008), pp. 318–338. ISSN: 0003-3022. DOI: 10.1097/ALN.0b013e31817f6d76.

[29] M. Díaz and D. E. Becker. "Thermoregulation: physiological and clinical considerations during sedation and general anesthesia". In: *Anesthesia progress* 57.1 (2010), pp. 25–33. ISSN: 0003-3006. DOI: 10.2344/0003-3006-57.1.25.

[30] M. N. Diringer, N. L. Reaven, S. E. Funk, and G. C. Uman. "Elevated body temperature independently contributes to increased length of stay in neurologic intensive

care unit patients". In: *Critical care medicine* 32.7 (2004), pp. 1489–1495. ISSN: 0090-3493. DOI: 10.1097/01.CCM.0000129484.61912.84.

[31] A. Macario and F. Dexter. "What are the most important risk factors for a patient's developing intraoperative hypothermia?" In: *Anesthesia & Analgesia* 94.1 (2002), pp. 215–220. ISSN: 0003-2999. DOI: 10.1213/00000539-200201000-00042.

[32] S. M. Frank, L. A. Fleisher, M. J. Breslow, M. S. Higgins, K. F. Olson, S. Kelly, and C. Beattie. "Perioperative maintenance of normothermia reduces the incidence of morbid cardiac events: a randomized clinical trial". In: *Journal of the American Medical Association* 277 (1997), pp. 1127–1134. ISSN: 0098-7484. DOI: 10.1001/jama.1997.03540380041029.

[33] S. Rajagopalan, E. Mascha, J. Na, and D. I. Sessler. "The effects of mild perioperative hypothermia on blood loss and transfusion requirement". In: *Anesthesiology* 108.1 (2008), pp. 71–77. ISSN: 0003-3022. DOI: 10.1097/01.anes.0000296719.73450.52.

[34] B. Romlin, K. Petruson, and K. Nilsson. "Moderate superficial hypothermia prolongs bleeding time in humans". In: *Acta anaesthesiologica scandinavica* 51.2 (2007), pp. 198–201. ISSN: 0001-5172. DOI: 10.1111/j.1399-6576.2006.01181.x.

[35] T. Heier, J. E. Caldwell, D. I. Sessler, and R. D. Miller. "Mild intraoperative hypothermia increases duration of action and spontaneous recovery of vecuronium blockade during nitrous oxide-isoflurane anesthesia in humans". In: *Anesthesiology* 74.5 (1991), pp. 815–819. ISSN: 0003-3022.

[36] A. C. Melling, B. Ali, E. M. Scott, and D. J. Leaper. "Effects of preoperative warming on the incidence of wound infection after clean surgery: a randomised controlled trial". In: *The Lancet* 358.9285 (2001), pp. 876–880. ISSN: 0140-6736. DOI: 10.1016/S0140-6736(01)06071-8.

[37] E. M. Scott and R. Buckland. "A systematic review of intraoperative warming to prevent postoperative complications". In: *AORN journal* 83.5 (2006), pp. 1090–1113. ISSN: 0001-2092. DOI: 10.1016/S0001-2092(06)60120-8.

[38] P. Kan, E. Duckworth, A. V. Germanwala, P. Pelargos, J. M. Cho, W. Choy, Z. A. Smith, and I. Yang. "Neurosurgery concepts: Key perspectives on embolectomy for stroke with emergent large vessel occlusion (MR CLEAN), endonasal endoscopic craniopharyngioma resection, gamma knife radiosurgery for meningiomas, therapeutic hypothermia for severe traumatic brain injury". In: *Surgical Neurology International* 6.165 (2015). ISSN: 2152-7806. DOI: 10.4103/2152-7806.168064.

[39] DrugBank Database. URL: https://www.drugbank.ca (visited on 2017-06-14).

[40] *Flüssigkeits- und Volumentherapie - Jonosteril™ Infusionslösung*. manufacturer's datasheet. Fresenius Kabi Deutschland GmbH. 2015.

[41] N. H. Franco and I. A. S. Olsson. "Scientists and the 3Rs: attitudes to animal use in biomedical research and the effect of mandatory training in laboratory animal

science". In: *Laboratory animals* 48.1 (2014), pp. 50–60. ISSN: 0023-6772. DOI: 10.1177/0023677213498717.

[42] W. M. S. Russell, R. L. Burch, and C. W. Hume. "The principles of humane experimental technique". In: (1959).

[43] I. A. S. Olsson, N. H. Franco, D. M. Weary, and P. Sandøe. "The 3Rs principle–mind the ethical gap!" In: *Alternatives and Animal Use in the Life (ALTEX), Proceedings of the 8th World Congress on*. Johns Hopkins University Press, 2012, pp. 333–336.

[44] *DIRECTIVE 2010/63/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 22 September 2010 on the protection of animals used for scientific purposes*. 2010. URL: http://eur-lex.europa.eu/eli/dir/2010/63/oj (visited on 2017-09-17).

[45] A. Bassols, C. Costa, P. D. Eckersall, J. Osada, J. Sabria, and J. Tibau. "The pig as an animal model for human pathologies: A proteomics perspective". In: *Proteomics - Clinical Applications* 8.9–10 (2014), pp. 715–731. DOI: 10.1002/prca.201300099.

[46] E. Kobayashi, S. Hishikawa, T. Teratani, and A. T. Lefor. "The pig as a model for translational research: overview of porcine animal models at Jichi Medical University". In: *Transplantation Research* 1.1 (2012). DOI: 10.1186/2047-1440-1-8.

[47] T. P. Sullivan, W. H. Eaglstein, S. C. Davis, and P. Mertz. "The pig as a model for human wound healing". In: *Wound Repair and Regeneration* 9.2 (2001), pp. 66–76. DOI: 10.1046/j.1524-475x.2001.00066.x.

[48] J. K. Lunney. "Advances in Swine Biomedical Model Genomics". In: *International Journal of Biological Sciences* 3.3 (2007), pp. 179–184. ISSN: 1449-2288. DOI: 10.7150/ijbs.3.179.

[49] M. Swindle, A. Makin, A. Herron, F. Clubb Jr, and K. Frazier. "Swine as models in biomedical research and toxicology testing". In: *Veterinary Pathology* 49.2 (2012), pp. 344–356. DOI: 10.1177/0300985811402846.

[50] M. M. Swindle. "Swine as Surgical Models in Biomedical Research". In: *Proceedings of the ACVP/ASVCP Concurrent Annual Meetings*. 2009.

[51] M. M. Swindle and A. C. Smith. *Swine in the laboratory: surgery, anesthesia, imaging, and experimental techniques*. CRC Press, 2015. ISBN: 9781466553477.

[52] J. Litten-Brown, A. Corson, and L. Clarke. "Porcine models for the metabolic syndrome, digestive and bone disorders: a general overview". In: *Animal* 4.06 (2010), pp. 899–920. ISSN: 1751-732X. DOI: 10.1017/S1751731110000200.

[53] M. Kixmöller. "Labordiagnostische Referenzbereiche bei unterschiedlichen Schweinerassen sowie histopathologische und immunhistochemische Untersuchung von Gehirnen älterer Sauen und Eber auf transmissible spongiforme Enzephalopathie im Rahmen der TSE-Studie". urn:nbn:de:bvb:19-21986. doc-

toral thesis. München, Germany: Tierärztlichen Fakultät der Ludwig-Maximilians-Universität München, 2004. URL: http://d-nb.info/971613702.

[54] E. Nerbas. “Aktualisierung von Blutparametern beim Schwein”. doctoral thesis. Hannover, Germany: Klinik für kleine Klauentiere und forensische Medizin und Ambulatorische Klinik der Tierärztliche Hochschule Hannover, 2008. URL: http://d-nb.info/991973186.

[55] L. Thomas and S. Abdelhamid. *Labor und Diagnose: Indikation und Bewertung von Laborbefunden für die medizinische Diagnostik*. Die Medizinische Verlagsgesellschaft, 1992. ISBN: 9783921320211.

[56] P. J. Bollen, A. K. Hansen, and H. J. Rasmussen. *The Laboratory Swine*. Laboratory Animal Pocket Reference. CRC Press, 2000. ISBN: 0849310350.

[57] W. Harris. “Hemoglobin, blood gases and serum electrolyte values in swine”. In: *The Canadian Veterinary Journal* 15.10 (1974), p. 282. ISSN: 0008-5286.

[58] E. Glawischnig, K. Schlerker, W. Schuller, and W. Baumgartner. *Arbeitswerte in der Laboratoriumsdiagnostik beim Schwein*. Tierärztlicher Mitschrieb. 64. Wien, Austria, 1977, pp. 341–346.

[59] *Appendixes*. MERCK Veterenary Manual. URL: http://www.msdvetmanual.com/appendixes (visited on 2017-09-17).

[60] H. Foundation. *Guideline for diagnosis and management of hypertension in adults*. 2017. URL: https://www.heartfoundation.org.au/images/uploads/publications/PRO-167_Hypertension-guideline-2016_WEB.pdf (visited on 2017-09-28).

[61] J. W. Mason, D. J. Ramseth, D. O. Chanter, T. E. Moon, D. B. Goodman, and B. Mendzelevski. “Electrocardiographic reference ranges derived from 79,743 ambulatory subjects”. In: *Journal of Electrocardiology* 40.3 (2007), pp. 228–234. ISSN: 0022-0736. DOI: 10.1016/j.jelectrocard.2006.09.003.

[62] K. Barrett, S. Barman, S. Boitano, and H. Brooks. *Ganong’s Review of Medical Physiology*. 24th ed. McGraw-Hill, 2012, p. 619. ISBN: 9780071780032.

[63] Z. Su, J. Oto, J. Wang, W. R. Kimball, C. T. Chenelle, R. M. Kacmarek, D. R. King, Y. Jiang, and M. J. Duggan. “Validation of Respiratory Inductance Plethysmography for Measuring Tidal Volume in Swine”. In: *Comparative medicine* 65.3 (2015), pp. 225–231. ISSN: 1532-0820.

[64] R. Renner and M. Haller. “Wasser- und Elektrolythaushalt”. In: R. Rossaint, C. Werner, and B. Zwießler. *Die Anästhesiologie*. 3rd ed. .8. Springer, 2012. ISBN: 978-3-642-21125-6.

[65] P. Kielstein and E. Wolfarth. *Schweinekrankheiten. Ätiologie - Pathogenese - Klinik - Therapie - Bekämpfung*. 3rd ed. Stuttgart, Germany: Ferdinand Enke Verlag, 1987. ISBN: 978-3432896335.

[66] *Request for Comments - Dynamic Host Configuration Protocol (RFC2131)*. Network Working Group, 1997-04. URL: https://tools.ietf.org/html/rfc2131 (visited on 2017-06-09).

[67] *Request for Comments - Domain Names: Concepts and Facilities (RFC1034)*. Network Working Group, 1987-11. URL: https://tools.ietf.org/html/rfc1034 (visited on 2017-06-09).

[68] *Request for Comments - Domain Names: Implementation and Specifications (RFC1035)*. Network Working Group, 1987-11. URL: https://tools.ietf.org/html/rfc1035 (visited on 2017-06-09).

[69] *Request for Comments - Clarifications to the DNS Specification (RFC2181)*. Network Working Group, 1997-07. URL: https://tools.ietf.org/html/rfc2181 (visited on 2017-06-09).

[70] *Request for Comments - Network Time Protocol Version 4: Protocol and Algorithms Specification (RFC5905)*. Network Working Group, 2010-10. URL: https://tools.ietf.org/html/rfc5905 (visited on 2017-06-09).

[71] *IEEE Standard 802.1Q-2014 – Bridges and Bridged Networks*. IEEE Standard for Local and metropolitan area networks, 2014. URL: http://standards.ieee.org/getieee802/download/802-1Q-2014.pdf (visited on 2017-06-08).

[72] *IEEE Standard 802.1AX-2014 – Link Aggregation*. IEEE Standard for Local and metropolitan area networks, 2014. URL: http://standards.ieee.org/getieee802/download/802.1AX-2014.pdf (visited on 2017-06-08).

[73] *IEEE Standard 802.1D-2004 – Media Access Control (MAC) Bridges*. IEEE Standard for Local and metropolitan area networks, 2004. URL: http://standards.ieee.org/getieee802/download/802.1D-2004.pdf (visited on 2017-06-08).

[74] *Qt – Cross-platform software development for embedded & desktop*. Qt. URL: https://www.qt.io/ (visited on 2017-06-27).

[75] *Qt Documentation - Signals and Slots*. Qt. 2017. URL: http://doc.qt.io/qt-5/signalsandslots.html (visited on 2017-06-09).

[76] The R Project for Statistical Computing. URL: https://www.r-project.org (visited on 2017-06-08).

[77] *About R*. The R Project for Statistical Computing. URL: https://www.r-project.org/about.html (visited on 2017-06-08).

[78] The HDF Group. URL: https://www.hdfgroup.org (visited on 2017-06-08).

[79] *HDF5*. The HDF Group. URL: https://www.hdfgroup.org/hdf5/.

[80] *Support: HDF5 Technologies*. The HDF Group. URL: https://support.hdfgroup.org/about/hdf%5C_technologies.html (visited on 2017-06-08).

[81] *MATLAB Documentation: HDF5 Files*. Version R2017a. MathWorks. URL: https://de.mathworks.com/help/matlab/hdf5-files.html (visited on 2017-06-08).

[82] M. Smith. *HDF5 interface to R*. Version 3.5. Bioconductor. 2017. URL: http://bioconductor.org/packages/release/bioc/html/rhdf5.html (visited on 2017-06-08).

[83] B. Fischer and G. Pau. *rhdf5: HDF5 interface to R*. Version R package version 2.20.0. 2017.

[84] Health Level Seven International. URL: https://www.hl7.org/ (visited on 2017-06-09).

[85] *HL7 Version 2.x standards*. Health Level Seven International. URL: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185 (visited on 2017-06-09).

[86] *HL7 Version 3.x standards*. Health Level Seven International. URL: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=186 (visited on 2017-06-09).

[87] *PostgreSQL*. The PostgreSQL Global Development Group. URL: https://www.postgresql.org/ (visited on 2017-06-09).

[88] T. Studer. *Relationale Datenbanken: Von den theoretischen Grundlagen zu Anwendungen mit PostgreSQL*. eXamen.press. Springer, 2016. ISBN: 9783662465714.

[89] "Dealing with Missing or Incomplete Data: Debunking the Myth of Emptiness". In: J. W. Osborne. *Best Practices in Data Cleaning: A Complete Guide to Everything You Need to Do Before and After Collecting Your Data*. .6. SAGE, 2013, pp. 105–138. ISBN: 9781412988018. DOI: 10.4135/9781452269948.n6.

[90] R. J. Little and D. B. Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2002. ISBN: 978-0-471-18386-0.

[91] L. A. Zadeh. "Fuzzy sets". In: *Information and control* 8.3 (1965), pp. 338–353. ISSN: 0019-9958. DOI: 10.1016/S0019-9958(65)90241-X.

[92] L. A. Zadeh. "Fuzzy algorithms". In: *Information and control* 12.2 (1968), pp. 94–102. ISSN: 0019-9958. DOI: 0.1016/S0019-9958(68)90211-8.

[93] L. A. Zadeh. "Fuzzy logic". In: *Computer* 21.4 (1988), pp. 83–93. ISSN: 0018-9162. DOI: 10.1109/2.53.

[94] L. A. Zadeh. "Fuzzy logic= computing with words". In: *IEEE transactions on fuzzy systems* 4.2 (1996), pp. 103–111. ISSN: 1063-6706. DOI: 10.1109/91.493904.

[95] L. A. Zadeh. "The information principle". In: *Information Sciences* 294 (2015), pp. 540–549. ISSN: 0020-0255. DOI: 10.1016/j.ins.2014.09.026.

[96] H. Singh, M. M. Gupta, T. Meitzler, Z.-G. Hou, K. K. Garg, A. M. Solo, and L. A. Zadeh. "Real-life applications of fuzzy logic". In: *Advances in Fuzzy Systems* 2013 (2013). ISSN: 1687-711X. DOI: 10.1155/2013/581879.

[97] E. H. Mamdani. "Application of fuzzy algorithms for control of simple dynamic plant". In: *Proceedings of the Institution of Electrical Engineers*. Vol. 121. 12. IET. 1974, pp. 1585–1588. DOI: 10.1049/piee.1974.0328.

[98] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. 2nd ed. Springer, 2009. ISBN: 978-0-387-84857-0. DOI: 10.1007/978-0-387-84858-7.

[99] R. Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 58.1 (1996), pp. 267–288.

[100] H. Zou and T. Hastie. "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320. DOI: 10.1111/j.1467-9868.2005.00503.x.

[101] P. W. Holland and R. E. Welsch. "Robust regression using iteratively reweighted least-squares". In: *Communications in Statistics – Theory and Methods* 6.9 (1977), pp. 813–827. ISSN: 0361-0926. DOI: 10.1080/03610927708827533.

[102] J. O. Street, R. J. Carroll, and D. Ruppert. "A note on computing robust regression estimates via iteratively reweighted least squares". In: *The American Statistician* 42.2 (1988), pp. 152–154. ISSN: 0003-1305. DOI: 10.2307/2684491.

[103] A. J. Smola and B. Schölkopf. "A tutorial on support vector regression". In: *Statistics and computing* 14.3 (2004), pp. 199–222. ISSN: 0960-3174. DOI: 10.1023/B:STCO.0000035301.49549.88.

[104] D. Basak, S. Pal, and D. C. Patranabis. "Support vector regression". In: *Neural Information Processing-Letters and Reviews* 11.10 (2007), pp. 203–224. ISSN: 1738-2572.

[105] J. Platt. *Sequential minimal optimization: A fast algorithm for training support vector machines*. Tech. rep. Microsoft Research Lab - Redmond, 1998.

[106] M. Saeed, M. Villarroel, A. T. Reisner, G. Clifford, L.-W. Lehman, G. Moody, T. Heldt, T. H. Kyaw, B. Moody, and R. G. Mark. "Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): a public-access intensive care unit database". In: *Critical care medicine* 39.5 (2011), p. 952. ISSN: 0090-3493. DOI: 10.1097/CCM.0b013e31820a92c6.

[107] D. J. Scott, J. Lee, I. Silva, S. Park, G. B. Moody, L. A. Celi, and R. G. Mark. "Accessing the public MIMIC-II intensive care relational database for clinical research". In: *BMC Medical Informatics and Decision Making* 13.1 (2013), p. 9. ISSN: 1472-6947. DOI: 10.1186/1472-6947-13-9.

[108] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. "MIMIC-III, a freely accessible critical care database". In: *Scientific Data* 3 (2016). ISSN: 2052-4463. DOI: 10.1038/sdata.2016.35.

[109] P. R. Norris and B. M. Dawant. "Closing the loop in ICU decision support: physiologic event detection, alerts, and documentation". In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association. 2001, pp. 498–502. DOI: 10.1197/jamia.M1238.

[110] S. Pöhlsen, S. Schlichting, M. Strähle, F. Franz, and C. Werner. "A concept for a medical device plug-and-play architecture based on web services". In: *SIGBED Review* 6.2 (2009), 6:1–6:7. ISSN: 1551-3688. DOI: 10.1145/1859823.1859829.

[111] M. Feng, Z. Zhang, F. Zhang, Y. Ge, L. Y. Loy, K. Vellaisamy, W. Guo, P. L. Chin, N. K. K. King, B. T. Ang, et al. "iSyNCC: An intelligent system for patient monitoring & clinical decision support in neuro-critical-care". In: *Engineering in Medicine and Biology Society (EMBC), 2011 Annual International Conference of the IEEE*. IEEE. 2011, pp. 6426–6429. ISBN: 978-1-4244-4122-8. DOI: 10.1109/IEMBS.2011.6091586.

[112] T. Heinze, R. Wierschke, A. Schacht, and M. von Löwis. "A hybrid artificial intelligence system for assistance in remote monitoring of heart patients". In: *International Conference on Hybrid Artificial Intelligence Systems*. Vol. 6679. Springer. 2011, pp. 413–420. DOI: 10.1007/978-3-642-21222-2_50.

[113] S. Franke and T. Neumuth. "Rule-based medical device adaptation for the digital operating room". In: *Engineering in Medicine and Biology Society (EMBC), 37th Annual International Conference of*. IEEE. 2015, pp. 1733–1736. ISBN: 978-1-4244-9271-8. DOI: 10.1109/EMBC.2015.7318712.

[114] M. Kasparick, S. Schlichting, F. Golatowski, and D. Timmermann. "New IEEE 11073 standards for interoperable, networked point-of-care medical devices". In: *Engineering in Medicine and Biology Society (EMBC), 37th Annual International Conference of the IEEE*. IEEE. 2015, pp. 1721–1724. DOI: 10.1109/EMBC.2015.7318709.

[115] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. "ROS: an open-source Robot Operating System". In: *ICRA workshop on open source software*. Vol. 3. 3.2. 2009, p. 5.

[116] R. Allen and D. Smith. "Neuro-fuzzy closed-loop control of depth of anaesthesia". In: *Artificial Intelligence in Medicine* 21.1–3 (2001), pp. 185–191. ISSN: 0933-3657. DOI: 10.1016/S0933-3657(00)00084-1.

[117] G. Clifford, W. Long, G. Moody, and P. Szolovits. "Robust parameter extraction for decision support using multimodal intensive care data". In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 367.1887 (2009), pp. 411–429. ISSN: 1364-503X. DOI: 10.1098/rsta.2008.0157.

[118] A. E. Johnson, A. A. Kramer, and G. D. Clifford. "Data preprocessing and mortality prediction: The Physionet/CinC 2012 challenge revisited". In: *Computing in Cardiology Conference (CinC), 2014*. IEEE. 2014, pp. 157–160. ISBN: 978-1-4799-4347-0.

[119] A. Mäkivirta, E. Koski, A. Kari, and T. Sukuvaara. "The median filter as a preprocessor for a patient monitor limit alarm system in intensive care". In: *Computer methods and programs in biomedicine* 34.2-3 (1991), pp. 139–144. ISSN: 0169-2607. DOI: 10.1016/0169-2607(91)90039-V.

[120] M. Borowski, S. Siebig, C. Wrede, and M. Imhoff. “Reducing false alarms of intensive care online-monitoring systems: an evaluation of two signal extraction algorithms”. In: *Computational and Mathematical Methods in Medicine* 2011 (2011). ISSN: 1748-6718. DOI: 10.1155/2011/143480.

[121] E. J. Knobbe and B. Buckingham. “The extended Kalman filter for continuous glucose monitoring”. In: *Diabetes technology & therapeutics* 7.1 (2005), pp. 15–27. ISSN: 1520-9156. DOI: 10.1089/dia.2005.7.15.

[122] Q. Li, R. G. Mark, and G. D. Clifford. “Robust heart rate estimation from multiple asynchronous noisy sources using signal quality indices and a Kalman filter”. In: *Physiological measurement* 29.1 (2008), p. 15. ISSN: 0967-3334.

[123] P. Yang, G. A. Dumont, and J. M. Ansermino. “Sensor fusion using a hybrid median filter for artifact removal in intraoperative heart rate monitoring”. In: *Journal of Clinical Monitoring and Computing* 23.2 (2009), pp. 75–83. ISSN: 1387-1307. DOI: 10.1007/s10877-009-9163-2.

[124] C. Williams, J. Quinn, and N. McIntosh. “Factorial switching Kalman filters for condition monitoring in neonatal intensive care”. In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 18. MIT Press, 2006, pp. 1513–1520. ISBN: 9780262232531.

[125] F. Steimann and K.-P. Adlassnig. “Two-stage interpretation of ICU data based on fuzzy sets”. In: *Working Papers of AAAI Spring Symposium Series: Interpreting Clinical Data*. Stanford University. 1994, pp. 146–150.

[126] I. J. Haimowitz, P. P. Le, and I. S. Kohane. “Clinical monitoring using regression-based trend templates”. In: *Artificial intelligence in medicine* 7.6 (1995), pp. 473–496. ISSN: 0933-3657. DOI: 10.1016/0933-3657(95)00023-6.

[127] K. Schettlinger, R. Fried, and U. Gather. “Robust filters for intensive care monitoring: beyond the running median/Robuste Filter für intensivmedizinisches Monitoring: mehr als ein gleitender Median”. In: *Biomedizinische Technik* 51.2 (2006), pp. 49–56. ISSN: 1862-278X. DOI: 10.1515/BMT.2006.010.

[128] M. Imhoff, M. Bauer, U. Gather, and D. Löhlein. “Statistical pattern detection in univariate time series of intensive care on-line monitoring data”. In: *Intensive Care Medicine* 24.12 (1998), pp. 1305–1314. ISSN: 0342-4642. DOI: 10.1007/s001340050767.

[129] R. Fried, U. Gather, and M. Imhoff. “Online pattern recognition in intensive care medicine.” In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association (AMIA). 2001, pp. 184–188.

[130] M. Bauer, U. Gather, and M. Imhoff. *The identification of multiple outliers in online monitoring data*. Tech. rep. Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund, 1999.

[131] D. Sow, A. Biem, J. Sun, J. Hu, and S. Ebadollahi. “Real-time prognosis of ICU physiological data streams”. In: *Engineering in Medicine and Biology Society*

(EMBC), 2010 Annual International Conference of. IEEE. 2010, pp. 6785–6788. DOI: 10.1109/IEMBS.2010.5625983.

[132] A. Aboukhalil, L. Nielsen, M. Saeed, R. G. Mark, and G. D. Clifford. "Reducing false alarm rates for critical arrhythmias using the arterial blood pressure waveform". In: *Journal of Biomedical Informatics* 41.3 (2008), pp. 442–451. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2008.03.003.

[133] I. Achour, K. Nouira, and A. Trabelsi. "Reducing False Alarms in Intensive Care Units Based on Wavelets Technology". In: *International Journal of Bio-Science and Bio-Technology* 4.2 (2012), pp. 111–120. ISSN: 2233-7849.

[134] F. Wadehn, D. Carnal, and H.-A. Loeliger. "Estimation of heart rate and heart rate variability from pulse oximeter recordings using localized model fitting". In: *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. IEEE. 2015, pp. 3815–3818. DOI: 10.1109/EMBC.2015.7319225.

[135] P. Lal, C. K. I. Williams, K. Georgatzis, C. Hawthorne, P. McMonagle, I. Piper, and M. Shaw. "Detecting Artifactual Events in Vital Signs Monitoring Data". In: *Machine Learning for Healthcare Technologies*. Ed. by D. A. Clifton. IET, 2016-10, pp. 7–32. ISBN: 978-1-84919-978-0. DOI: 10.1049/PBHE002E_ch2.

[136] J. Rinehart, N. Liu, B. Alexander, and M. Cannesson. "Closed-loop systems in anesthesia: Is there a potential for closed-loop fluid management and hemodynamic optimization?" In: *Anesthesia & Analgesia* 114.1 (2012), pp. 130–143. ISSN: 0003-2999. DOI: 10.1213/ANE.0b013e318230e9e0.

[137] J. H. Bates and M. P. Young. "Applying Fuzzy Logic to Medical Decision Making in the Intensive Care Unit". In: *American Journal of Respiratory and Critical Care Medicine* 167.7 (2003), pp. 948–952. ISSN: 1073-449X. DOI: 10.1164/rccm.200207-777CP.

[138] C. R. Leite, G. R. Sizilio, A. D. Neto, R. A. Valentim, and A. M. Guerreiro. "A fuzzy model for processing and monitoring vital signs in ICU patients". In: *Biomedical engineering online* 10.1 (2011), p. 68. ISSN: 1475-925X. DOI: 10.1186/1475-925X-10-68.

[139] E. Papageorgiou. "Medical decision making through fuzzy computational intelligent approaches". In: *Foundations of Intelligent Systems* (2009), pp. 99–108. ISSN: 0302-9743. DOI: 10.1007/978-3-642-04125-9_13.

[140] M. M. Baig, H. GholamHosseini, A. Kouzani, and M. J. Harrison. "Anaesthesia monitoring using fuzzy logic". In: *Journal of clinical monitoring and computing* 25.5 (2011), p. 339. ISSN: 1387-1307. DOI: 10.1007/s10877-011-9315-z.

[141] C. Vaity, N. Al-Subaie, and M. Cecconi. "Cooling techniques for targeted temperature management post-cardiac arrest". In: *Critical Care* 19.1 (2015), p. 103. ISSN: 1364-8535. DOI: 10.1186/s13054-015-0804-1.

[142] T. C. Mort, T. D. Rintel, and F. Altman. “The effects of forced-air warming on postbypass central and skin temperatures and shivering activity”. In: *Journal of Clinical Anesthesia* 8.5 (1996), pp. 361–370. ISSN: 0952-8180. DOI: 10.1016/0952-8180(96)00081-5.

[143] O. J. Campastro, S. A. Gonzalez, and J. H. Pazo. “Automatic temperature controller for maintaining body temperature in experimental animals”. In: *Journal of Neuroscience Methods* 30.3 (1989), pp. 185–187. ISSN: 0165-0270.

[144] H. H. Qiu, G. P. Cofer, L. W. Hedlund, and G. A. Johnson. “Automated feedback control of body temperature for small animal studies with MR microscopy”. In: *IEEE Transactions on Biomedical Engineering* 44.11 (1997), pp. 1107–1113. ISSN: 0018-9294. DOI: 10.1109/10.641338.

[145] H. I. Kim, H. C. Kim, and B. W. Yoon. “Automatic body temperature control system for small animal studies using dual mode PI control”. In: *20th Annual International Conference of the Engineering in Medicine and Biology Society, Proceedings of*. Vol. 4. IEEE, pp. 1967–1969. DOI: 10.1109/IEMBS.1998.746987.

[146] P. Britto, S. Veerachamy, J. R. Yathav, and H. K. Rahman. “Automated Infusion Control with Core Body Temperature for Infants Under IV Administration”. In: *7th WACBE World Congress on Bioengineering*. Vol. 52. Springer. 2015, pp. 118–121. DOI: 10.1007/978-3-319-19452-3_32.

[147] R. Rattan and S. A. Nasraway. “The future is now: software-guided intensive insulin therapy in the critically ill”. In: *Journal of diabetes science and technology* 7.2 (2013), pp. 548–554. ISSN: 1932-2968. DOI: 10.1177/193229681300700231.

[148] B. W. Bequette. “Continuous glucose monitoring: real-time algorithms for calibration, filtering, and alarms”. In: *Journal of Diabetes Science and Technology* 4.2 (2010), pp. 404–418. ISSN: 1932-2968. DOI: 10.1177/193229681000400222.

[149] R. Hovorka. “Continuous glucose monitoring and closed-loop systems”. In: *Diabetic Medicine* 23.1 (2006), pp. 1–12. ISSN: 0742-3071. DOI: 10.1111/j.1464-5491.2005.01672.x.

[150] B. T. Thompson, J. F. Orme, H. Zheng, P. M. Luckett, J. D. Truwit, D. F. Willson, R. D. Hite, R. G. Brower, G. R. Bernard, M. A. Curley, J. S. Steingrub, D. K. Sorenson, K. Sward, E. Hirshberg, and A. H. Morris. “Multicenter validation of a computer-based clinical decision support tool for glucose control in adult and pediatric intensive care units”. In: *Journal of Diabetes Science and Technology* 2.3 (2008), pp. 357–268. ISSN: 1932-2968. DOI: 10.1177/193229680800200304.

[151] J. Plank, J. Blaha, J. Cordingley, M. E. Wilinska, L. J. Chassin, C. Morgan, S. Squire, M. Haluzik, J. Kremen, S. Svacina, et al. “Multicentric, randomized, controlled trial to evaluate blood glucose control by the model predictive control algorithm versus routine glucose management protocols in intensive care unit patients”. In: *Diabetes Care* 29.2 (2006), pp. 271–276. ISSN: 0149-5992. DOI: 10.2337/diacare.29.02.06.dc05-1689.

[152] C. Pachler, J. Plank, H. Weinhandl, R. Hovorka, L. Chassin, P. Kaufmann, K. Smolle, T. Pieber, and M. Ellmerer. "Evaluation of a model predictive control algorithm using time-variant sampling to establish tight glycaemic control in clinical practice". In: *Critical Care* 11.Suppl 2 (2007), P138. ISSN: 1364-8535. DOI: 10.1186/cc5298.

[153] C. Pachler, J. Plank, H. Weinhandl, L. J. Chassin, M. E. Wilinska, R. Kulnik, P. Kaufmann, K.-H. Smolle, E. Pilger, T. R. Pieber, et al. "Tight glycaemic control by an automated algorithm with time-variant sampling in medical ICU patients". In: *Intensive Care Medicine* 34.7 (2008), pp. 1224–1230. ISSN: 0342-4642. DOI: 10.1007/s00134-008-1033-8.

[154] R. Kulnik, J. Plank, C. Pachler, M. E. Wilinska, A. Groselj-Strele, D. Röthlein, M. Wufka, N. Kachel, K. H. Smolle, S. Perl, T. R. Pieber, R. Hovorka, and M. Ellmerer. "Evaluation of implementation of a fully automated algorithm (enhanced model predictive control) in an interacting infusion pump system for establishment of tight glycemic control in medical intensive care unit patients". In: *Journal of Diabetes Science and Technology* 2.6 (2008), pp. 963–970. ISSN: 1932-2968. DOI: 10.1177/193229680800200606.

[155] *B. Braun Space GlucoseControl*. Accessed: 2017-08-29. URL: https://www.bbraun.de/de/products/b0/b-braun-space-glucosecontrol.html.

[156] P. C. Davidson, R. D. Steed, and B. W. Bode. "Glucommander: a computer-directed intravenous insulin system shown to be safe, simple, and effective in 120,618 h of operation". In: *Diabetes Care* 28.10 (2005), pp. 2418–2423. ISSN: 0149-5992. DOI: 10.2337/diacare.28.10.2418.

[157] B. W. Bequette. "Challenges and recent progress in the development of a closed-loop artificial pancreas". In: *Annual reviews in control* 36.2 (2012), pp. 255–266. ISSN: 1367-5788. DOI: 10.1016/j.arcontrol.2012.09.007.

[158] M. Krol and D. L. Reich. "The algorithm for detecting critical conditions during anesthesia". In: *Computer-Based Medical Systems (CBMS). Proceedings. 12th IEEE Symposium on*. IEEE. 1999, pp. 208–213. DOI: 10.1109/CBMS.1999.781272.

[159] M. Luginbühl, C. Bieniok, D. Leibundgut, R. Wymann, A. Gentilini, and T. W. Schnider. "Closed-loop Control of Mean Arterial Blood Pressure during Surgery with AlfentanilClinical Evaluation of a Novel Model-based Predictive Controller". In: *Anesthesiology* 105.3 (2006), pp. 462–470. ISSN: 0003-3022.

[160] J. Kizito. "Diagnesia: A prototype of a decision support system for anesthetists". In: *Broadband Communications, Information Technology & Biomedical Applications, 3rd Int. Conference on*. IEEE. 2008, pp. 12–19. ISBN: 978-1-4244-3281-3. DOI: 10.1109/BROADCOM.2008.39.

[161] W. M. Haddad and J. M. Bailey. "Closed-loop control for intensive care unit sedation". In: *Best Practice & Research Clinical Anaesthesiology* 23.1 (2009), pp. 95–114. ISSN: 1521-6896. DOI: 10.1016/j.bpa.2008.07.007.

[162] E. C. Borera, B. L. Moore, A. G. Doufas, and L. D. Pyeatt. “An adaptive neural network filter for improved patient state estimation in closed-loop anesthesia control”. In: *Tools with Artificial Intelligence (ICTAI), 23rd IEEE International Conference on*. IEEE. 2011, pp. 41–46. ISBN: 978-1-4577-2068-0. DOI: 10.1109/ICTAI.2011.15.

[163] S. Sviri, A. Bayya, P. D. Levin, R. Khalaila, I. Stav, and D. M. Linton. “Intelligent ventilation in the intensive care unit”. In: *Southern African Journal of Critical Care* 28.1 (2012), pp. 6–12. ISSN: 1562-8264. DOI: 10.7196/SAJCC.130.

[164] S. Ching, M. Y. Liberman, J. J. Chemali, M. B. Westover, J. D. Kenny, K. Solt, P. L. Purdon, and E. N. Brown. “Real-time Closed-loop Control in a Rodent Model of Medically Induced Coma Using Burst Suppression”. In: *Anesthesiology* 119.4 (2013), pp. 848–860. ISSN: 0003-3022. DOI: 10.1097/ALN.0b013e31829d4ab4.

[165] T. Hemmerling, E. Arbeid, M. Wehbe, S. Cyr, R. Taddei, and C. Zaouter. “Evaluation of a novel closed-loop total intravenous anaesthesia drug delivery system: a randomized controlled trial”. In: *British Journal of Anaesthesia* 110.6 (2013), pp. 1031–1039. ISSN: 0007-0912. DOI: 10.1093/bja/aet001.

[166] M. M. Shanechi, J. J. Chemali, M. Liberman, K. Solt, and E. N. Brown. “A brain-machine interface for control of medically-induced coma”. In: *PLoS computational biology* 9.10 (2013), e1003284. ISSN: 1553-734X. DOI: 10.1371/journal.pcbi.1003284.

[167] G. A. Dumont and J. M. Ansermino. “Closed-loop control of anesthesia: a primer for anesthesiologists”. In: *Anesthesia & Analgesia* 117.5 (2013), pp. 1130–1138. ISSN: 0003-2999. DOI: 10.1213/ANE.0b013e3182973687.

[168] M. Y. Liberman, S. Ching, J. Chemali, and E. N. Brown. “A closed-loop anesthetic delivery system for real-time control of burst suppression”. In: *Journal of Neural Engineering* 10.4 (2013), p. 046004. ISSN: 1741-2552. DOI: 10.1088/1741-2560/10/4/046004.

[169] W. M. Haddad, J. M. Bailey, B. Gholami, and A. R. Tannenbaum. “Clinical Decision Support and Closed-Loop Control for Intensive Care Unit Sedation”. In: *Asian Journal of Control* 15.2 (2013), pp. 317–339. ISSN: 1934-6093. DOI: 10.1002/asjc.701.

[170] W. Ngan Kee, K. Khaw, F. Ng, and Y. Tam. “Randomized comparison of closed-loop feedback computer-controlled with manual-controlled infusion of phenylephrine for maintaining arterial pressure during spinal anaesthesia for Caesarean delivery”. In: *British Journal of Anaesthesia* 110.1 (2013), pp. 59–65. ISSN: 0007-0912. DOI: 10.1093/bja/aes339.

[171] T. E. Miller and T. J. Gan. “Closed-loop systems in anesthesia: reality or fantasy?” In: *Anesthesia & Analgesia* 117.5 (2013), pp. 1039–1041. ISSN: 1526-7598. DOI: 10.1213/ANE.0b013e3182a5d689.

[172] M. D. Zilberberg, M. de Wit, J. R. Pirone, and A. F. Shorr. “Growth in adult prolonged acute mechanical ventilation: implications for healthcare delivery”. In: *Critical Care Medicine* 36.5 (2008), pp. 1451–1455. ISSN: 0090-3493. DOI: 10.1097/CCM.0b013e3181691a49.

[173] F. Lellouche, P.-A. Bouchard, S. Simard, E. L’Her, and M. Wysocki. “Evaluation of fully automated ventilation: a randomized controlled study in post-cardiac surgery patients”. In: *Intensive Care Medicine* 39.3 (2013), pp. 463–471. ISSN: 0342-4642. DOI: 10.1007/s00134-012-2799-2.

[174] T. Nemoto, G. E. Hatzakis, C. W. Thorpe, R. Olivenstein, S. Dial, and J. H. Bates. “Automatic control of pressure support mechanical ventilation using fuzzy logic”. In: *American Journal of Respiratory and Critical Care Medicine* 160.2 (1999), pp. 550–556. ISSN: 1073-449X. DOI: 10.1164/ajrccm.160.2.9809013.

[175] F. Tehrani, M. Rogers, T. Lo, T. Malinowski, S. Afuwape, M. Lum, B. Grundl, and M. Terry. “Closed-loop control of the inspired fraction of oxygen in mechanical ventilation”. In: *Journal of clinical monitoring and computing* 17.6 (2002), pp. 367–376. ISSN: 1387-1307. DOI: 10.1023/A:1024261021473.

[176] F. Tehrani, M. Rogers, T. Lo, T. Malinowski, S. Afuwape, M. Lum, B. Grundl, and M. Terry. “A dual closed-loop control system for mechanical ventilation”. In: *Journal of clinical monitoring and computing* 18.2 (2004), pp. 111–129. ISSN: 1387-1307. DOI: 10.1023/B:JOCM.0000032744.99885.38.

[177] F. T. Tehrani. “A new decision support system for mechanical ventilation”. In: *29th Annual International Conference Engineering in Medicine and Biology Society (EMBS), Proceedings of*. IEEE. 2007, pp. 3569–3572. DOI: 10.1109/IEMBS.2007.4353102.

[178] A. B. Otis, W. O. Fenn, and H. Rahn. “Mechanics of breathing in man”. In: *Journal of applied physiology* 2.11 (1950), pp. 592–607. ISSN: 0021-8987.

[179] C. G. Tams and N. R. Euliano. “Creating clinical decision support systems for respiratory medicine”. In: *37th Annual International Conference of Engineering in Medicine and Biology Society (EMBC), Proceedings of*. IEEE. 2015, pp. 5335–5338. ISBN: 978-1-4244-9271-8. DOI: 10.1109/EMBC.2015.7319596.

[180] M. J. Banner, N. R. Euliano, V. Brennan, C. Peters, A. J. Layon, and A. Gabrielli. “Power of breathing determined noninvasively with use of an artificial neural network in patients with respiratory failure”. In: *Critical Care Medicine* 34.4 (2006), pp. 1052–1059. ISSN: 0090-3493. DOI: 10.1097/01.CCM.0000206288.90613.1C.

[181] F. Lellouche, A. Bojmehrani, and K. Burns. “Mechanical ventilation with advanced closed-loop systems”. In: *New Developments in Mechanical Ventilation*. .16. European Respiratory Society, 2012. ISBN: 9781849840224. DOI: 10.1183/1025448x.10002911.

[182] E. Bialais, X. Wittebole, L. Vignaux, J. Roeseler, M. Wysocki, J. Meyer, G. Reychler, D. Novotni, T. Sottiaux, P.-F. Laterre, et al. "Closed-loop ventilation mode (IntelliVent®-ASV) in intensive care unit: a randomized trial." In: *Minerva anestesiologica* 82.6 (2016), pp. 657–668. ISSN: 0375-9393.

[183] Y.-C. Chu, C.-Z. Chen, C.-H. Lee, C.-W. Chen, H.-Y. Chang, and T.-R. Hsiue. "Prediction of arterial blood gas values from venous blood gas values in patients with acute respiratory failure receiving mechanical ventilation". In: *Journal of the Formosan Medical Association* 102.8 (2003), pp. 539–543. ISSN: 0929-6646.

[184] A. Ak, C. O. Ogun, A. Bayir, S. A. Kayis, and R. Koylu. "Prediction of arterial blood gas values from venous blood gas values in patients with acute exacerbation of chronic obstructive pulmonary disease". In: *The Tohoku Journal of Experimental Medicine* 210.4 (2006), pp. 285–290. ISSN: 0040-8727. DOI: doi.org/10.1620/tjem.210.285.

[185] K. Knubben, C. Thiel, M. Schenk, A. Etspüler, T. Schenk, M. H. Morgalla, and A. Königsrainer. "A new surgical model for hepatectomy in pigs". In: *European Surgical Research* 40.1 (2008), pp. 41–46. ISSN: 1421-9921. DOI: 10.1159/000108765.

[186] C. Thiel, K. Thiel, A. Etspüler, T. Schenk, M. H. Morgalla, A. Königsrainer, and M. Schenk. "Standardized intensive care unit management in an anhepatic pig model: new standards for analyzing liver support systems". In: *Critical Care* 14.4 (2010), R138. ISSN: 1364-8535. DOI: 10.1186/cc9196.

[187] K. Zwirner, C. Thiel, K. Thiel, M. H. Morgalla, A. Königsrainer, and M. Schenk. "Extracellular brain ammonia levels in association with arterial ammonia, intracranial pressure and the use of albumin dialysis devices in pigs with acute liver failure". In: *Metabolic Brain Disease* 25.4 (2010), pp. 407–412. ISSN: 1573-7365. DOI: 10.1007/s11011-010-9222-x.

[188] C. Thiel, K. Thiel, K. Scheuermann, E. Hawerkamp, A. Diewold, W. Klingert, J. Lauber, M. H. Morgalla, A. Königsrainer, and M. Schenk. "Acute liver failure by amanitin intoxication: Liver transplantation or wait for spontaneous regeneration? Evaluation of prognostic indicators in a porcine model". In: *Transplant International* 24 (2011), p. 10. ISSN: 1432-2277.

[189] C. Thiel, K. Thiel, A. Etspueler, M. Morgalla, S. Rubitschek, S. Schmid, W. Steurer, A. Königsrainer, and M. Schenk. "A reproducible porcine model of acute liver failure induced by intrajejunal acetaminophen administration". In: *European Surgical Research* 46.3 (2011), pp. 118–126. ISSN: 1421-9921. DOI: 10.1159/000323411.

[190] K. Thiel, W. Klingert, K. Klingert, M. H. Morgalla, M. U. Schuhmann, P. Leckie, Y. Sharifi, N. A. Davies, R. Jalan, A. Peter, C. Grasshoff, A. Königsrainer, M. Schenk, and C. Thiel. "Porcine model characterizing various parameters assessing the outcome after acetaminophen intoxication induced acute liver failure". In:

World Journal of Gastroenterology 23.9 (2017), pp. 1576–1585. ISSN: 2219-2840. DOI: 10.3748/wjg.v23.i9.1576.

[191] J. Peter, W. Klingert, A. Königsrainer, W. Rosenstiel, M. Bogdan, and M. Schenk. "TICoMS – A Modular and Message-Based Framework for Monitoring and Control of Medical Devices". In: *27th International Symposium on Computer-Based Medical Systems (CBMS), Proceedings of*. IEEE, 2014, pp. 473–474. ISBN: 978-1-4799-4435-4. DOI: 10.1109/CBMS.2014.96.

[192] Paessler PRTG. URL: https://www.paessler.com/prtg (visited on 2017-06-14).

[193] J. Peter, W. Klingert, M. Spüler, A. Königsrainer, W. Rosenstiel, and M. Schenk. "Automated therapeutic anticoagulation: A closed-loop approach using a modified measurement device". In: *Biomedical Engineering (BioMed), 13th IASTED International Conference on*. IEEE. 2017, pp. 224–228. ISBN: 978-0-88986-990-5. DOI: 10.2316/P.2017.852-027.

[194] R. A. Raschke, B. M. Reilly, J. R. Guidry, J. R. Fontana, and S. Srinivas. "The Weight-based Heparin Dosing Nomogram Compared with a Standard Care Nomogram - A Randomized Controlled Trial". In: *Annals of internal medicine* 119.9 (1993), pp. 874–881. ISSN: 0003-4819. DOI: 10.7326/0003-4819-119-9-199311010-00002.

[195] *Raspberry Pi website*. URL: https://www.raspberrypi.org/ (visited on 2017-07-16).

[196] *Wiring Pi – GPIO Interface library for the Raspberry Pi*. URL: http://wiringpi.com/ (visited on 2017-07-13).

[197] L. Enriquez and L. Shore-Lesserson. "Point-of-care coagulation testing and transfusion algorithms". In: *British Journal of Anaesthesia* 103.suppl_1 (2009), pp. i14–i22. ISSN: 0007-0912. DOI: 10.1093/bja/aep318.

[198] J. Peter, K. Klingert, W. Klingert, A. Königsrainer, C. Grasshoff, W. Rosenstiel, and M. Schenk. "Automated closed-loop management of body temperature using forced-air blankets: preliminary feasibility study in a porcine model". In: *BMC Anesthesiology* 18.80 (2018), pp. 1–11. ISSN: 1471-2253. DOI: 10.1186/s12871-018-0542-4.

[199] *3M Bair Hugger Model 755 Warming Unit Service Manual*. Accessed: 2017-09-17. URL: http://multimedia.3m.com/mws/media/798473O/model-775-service-manual-english.pdf.

[200] *Atmel Mega8A Mictrocontroller Summary Datasheet (Microchip Technology Inc.)* Accessed: 2017-09-17. URL: http://www.atmel.com/Images/Atmel-8159-8-bit-AVR-microcontroller-ATmega8A_summary.pdf.

[201] J. Rada-Vilela. *fuzzylite: a fuzzy logic control library*. 2017. URL: http://www.fuzzylite.com/.

[202] *IEEE Standard 1855-2016 – Standard for Fuzzy Markup Language*. IEEE Computational Intelligence Society, 2016.

[203] O. Wallin, J. Söderberg, B. Van Guelpen, H. Stenlund, K. Grankvist, and C. Brulin. "Blood sample collection and patient identification demand improvement: a questionnaire study of preanalytical practices in hospital wards and laboratories". In: *Scandinavian journal of caring sciences* 24.3 (2010), pp. 581–591. ISSN: 0283-9318. DOI: 10.1111/j.1471-6712.2009.00753.x.

[204] M. Plebani and P. Carraro. "Mistakes in a stat laboratory: types and frequency". In: *Clinical chemistry* 43.8 (1997), pp. 1348–1351. ISSN: 0009-9147.

[205] B. Aslan, J. Stemp, P. Yip, and J. Gun-Munro. "Method precision and frequent causes of errors observed in point-of-care glucose testing: a proficiency testing program perspective". In: *American Journal of Clinical Pathology* 142.6 (2014), pp. 857–863. ISSN: 0002-9173. DOI: 10.1309/AJCPP5YS2MVSKBYY.

[206] E. A. Wagar, L. Tamashiro, B. Yasin, L. Hilborne, and D. A. Bruckner. "Patient safety in the clinical laboratory: a longitudinal analysis of specimen identification errors". In: *Archives of pathology & laboratory medicine* 130.11 (2006), pp. 1662–1668. ISSN: 0003-9985.

[207] P. N. Valenstein, S. S. Raab, and M. K. Walsh. "Identification errors involving clinical laboratories: a College of American Pathologists Q-Probes study of patient and specimen identification errors at 120 institutions". In: *Archives of pathology & laboratory medicine* 130.8 (2006), pp. 1106–1113. ISSN: 0003-9985.

[208] P. A. Nutting, D. S. Main, P. M. Fischer, T. M. Stull, M. Pontious, M. Seifert, D. J. Boone, and S. Holcomb. "Problems in laboratory testing in primary care". In: *JAMA* 275.8 (1996), pp. 635–639. ISSN: 0098-7484. DOI: 10.1001/jama.1996.03530320059035.

[209] J. Ibojie and S. Urbaniak. "Comparing near misses with actual mistransfusion events: a more accurate reflection of transfusion errors". In: *British Journal of Haematology* 108.2 (2000), pp. 458–460. ISSN: 0007-1048. DOI: 10.1046/j.1365-2141.2000.01876.x.

[210] E. C. van Dongen-Lases, M. P. Cornes, K. Grankvist, M. Ibarz, G. B. Kristensen, G. Lippi, M. Nybo, A.-M. Simundic, et al. "Patient identification and tube labelling–a call for harmonisation". In: *Clinical Chemistry and Laboratory Medicine (CCLM)* 54.7 (2016), pp. 1141–1145. ISSN: 1434-6621. DOI: 10.1515/cclm-2015-1089.

[211] C. J. Huijsmans, F. G. Heilmann, A. G. Van Der Zanden, P. M. Schneeberger, and M. H. Hermans. "Single nucleotide polymorphism profiling assay to exclude serum sample mix-up". In: *VoxSanguinis* 92.2 (2007), pp. 148–153. ISSN: 1423-0410. DOI: 10.1111/j.1423-0410.2006.00871.x.

[212] R. Aron, S. Dutta, R. Janakiraman, and P. A. Pathak. "The impact of automation of systems on medical errors: evidence from field research". In: *Information Systems Research* 22.3 (2011), pp. 429–446. ISSN: 1526-5536.

[213] J. Peter, W. Klingert, K. Klingert, K. Thiel, D. Wulff, A. Königsrainer, W. Rosenstiel, and M. Schenk. “Algorithm-based arterial blood sampling recognition increasing safety in point-of-care diagnostics”. In: *World Journal of Critical Care Medicine* 6.3 (2017), pp. 172–178. ISSN: 2220-3141. DOI: 10.5492/wjccm.v6.i3.172.

[214] N. Aleks, S. J. Russell, M. G. Madden, D. Morabito, K. Staudenmayer, M. Cohen, and G. T. Manley. “Probabilistic detection of short events, with application to critical care monitoring”. In: *Advances in Neural Information Processing Systems (NIPS)*. Curran Associates, Inc., 2009, pp. 49–56.

[215] S. Haufe, F. Meinecke, K. Görgen, S. Dähne, J.-D. Haynes, B. Blankertz, and F. Bießmann. “On the interpretation of weight vectors of linear models in multivariate neuroimaging”. In: *NeuroImage* 87 (2014), pp. 96–110. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2013.10.067.

[216] R. S. Balaban. “The role of Ca^{2+} signaling in the coordination of mitochondrial ATP production with cardiac work”. In: *Biochimica et Biophysica Acta* 1787.11 (2009), pp. 1334–1341. ISSN: 0006-3002. DOI: 10.1016/j.bbabio.2009.05.011.

[217] S. Kojima, S. T. Wu, W. W. Parmley, and J. Wikman-Coffelt. “Relationship between intracellular calcium and oxygen consumption: effects of perfusion pressure, extracellular calcium, dobutamine, and nifedipine”. In: *American Heart Journal* 127.2 (1994), pp. 386–391. ISSN: 0002-8703. DOI: 10.1016/0002-8703(94)90129-5.

[218] S. K. Fellner, R. M. Lang, A. Neumann, K. T. Spencer, D. A. Bushinsky, and K. M. Borow. “Physiological mechanisms for calcium-induced changes in systemic arterial pressure in stable dialysis patients”. In: *Hypertension* 13.3 (1989), pp. 213–218. ISSN: 0194-911X. DOI: 10.1161/01.HYP.13.3.213.

[219] J. Van Nueten and P. Vanhoutte. “Improvement of tissue perfusion with inhibitors of calcium ion influx”. In: *Biochemical pharmacology* 29.4 (1980), pp. 479–481. ISSN: 0006-2952. DOI: 10.1016/0006-2952(80)90365-2.

[220] L. Chang, S. Roberts, and A. Welsh. “Robust Lasso Regression Using Tukey’s Biweight Criterion”. In: *Technometrics* (2017), pp. 1–12. ISSN: 0040-1706. DOI: 10.1080/00401706.2017.1305299.

[221] R. Pauldine, G. Beck, J. Salinas, and D. W. Kaczka. “Closed-loop strategies for patient care systems”. In: *Journal of Trauma and Acute Care Surgery* 64.4 (2008), S289–S294. ISSN: 0022-5282. DOI: 10.1097/TA.0b013e31816bce43.

[222] W. Klingert, J. Peter, C. Thiel, K. Thiel, W. Rosenstiel, K. Klingert, C. Grasshoff, A. Königsrainer, and M. Schenk. “Fully automated life support: An implementation and feasibility pilot study in healthy pigs”. In: *Intensive Care Medicine Experimental* 6.2 (2018), pp. 1–12. ISSN: 2197-425X. DOI: 10.1186/s40635-018-0168-3.