PhysioSim – A Full Hard- And Software Physiological Simulation Environment Applying A Hybrid Approach Based On Hierarchical Modeling Using Algebraic and Differential Systems and Dynamic Bayesian Networks

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Dipl. Inf. Ciamak Abkai
aus Teheran

Mannheim, 2010

Dekan: Professor Dr. Felix Freiling, Universität Mannheim
Referent: Professor Dr. Jürgen Hesser, Universität Heidelberg
Korreferent: Professor Dr. Reinhard Männer, Universität Heidelberg

Tag der mündlichen Prüfung: 08.10.2010

Abstract

A system for physiological modeling and simulation is presented. The architecture is considering hardware and software support for real-time physiological simulators, which are very important for medical education and risk management.

In contrary to other modeling methods, in this work the focus is to provide maximal modeling flexibility and extensibility. This is provided on the one hand by a hierarchical modeling notation in XML and on other hand by extending current methods by dynamic stochastic system modeling. Dynamic Bayesian Networks as well as deterministic system modeling by systems of algebraic and differential equations lead towards a sophisticated environment for medical simulation.

Specific simulations of haemodynamics and physiological based pharmocokinetics and pharmacodynamcis are performed by the proposed methods, demonstrating the applicability of the approaches.

In contrary to physiological modeling and analysis tools, for an educational simulator, the models have to be computed in real-time, which requires extensive design of the hardware and software architecture. For this purpose generic and extensible frameworks have been suggested and realized. All the components together lead to a novel physiological simulator environment, including a dummy, which emulates ECG, SaO2 and IBP vital signals in addition to software signal simulation.

The modeling approaches with DBN are furthermore analyzed in the domains of psychological and physiological reasoning, which should be integrated into a common basis for medical consideration.

Furthermore the system is used to show new concepts for dependable medical data monitoring, which are strongly related to physiological and psychological simulations.

Zusammenfassung

Ein vollständiges System zur physiologischen Modellierung und Simulation wird präsentiert. Meilensteine sind hierarchische Modellierung mit XML und Integration von deterministischen Systembeschreibungen durch Systeme von linearen algebraischen und differentiellen Gleichungen oder Kompartimenten und nicht-deterministische stochastische dynamische Systembeschreibungen durch Kombination von statischen und dynamischen Bayes'schen Netzen.

Die Systemarchitektur sieht eine dynamische Integration von Hardware (Emulations-Interfaces) und Software (zusätzliche physiologische Modelle) vor. Das System ist auf Echtzeit-Anwendung ausgelegt, da dies für die medizinische Ausbildung besonders wichtig ist. Spezielle Hardware-Schnittstellen (IBP, SaO2, ECG) wurden für die Emulation von quasi-echten Patienten-Vital-Parameter entwickelt, die eine flexible Erweiterung des medizinischen Simulators in Vergleich zur Full-Scale Simulatoren ermöglichen.

In einem detaillierten Stand der Technik werden die Vor- und Nachteile bisheriger industrieller und wissenschaftlicher Systeme und Verfahren erläutert und Ansätze zur Verbesserung aufgezeigt. Neue Verfahren und Methoden zur Modellierung und Simulation von physiologischen Systemen sowie Vitalfunktionen werden im Einzelnen vorgestellt und untersucht.

Anhand von Fallbeispielen (z.B. hämodynamische Simulation des Kardiovaskulären Systems) werden die modernen Verfahren auf ihrer Nützlichkeit im Einsatz des Systems für medizinische Simulationen untersucht und bewertet.

Zusätzlich werden Bayes'sche Netze bezüglich Simulation und Modellierung von kognitiven Modellen, die ebenfalls immer mehr an Bedeutung für physiologische Vorgänge gewinnen untersucht.

Abschließend wird das System zur Demonstration von neuen Konzepten zur Verbesserung der Verlässlichkeit von Physiologie-Monitor-Geräten eingesetzt, die ebenfalls - wie gezeigt wird - sehr eng mit physiologischen Simulationen von Patienten verknüpft sind.

University of Heidelberg
Medical Physics Group - Experimental Radiation Therapy

University of Mannheim

Thesis

PhysioSim - A Full Hard- And Software Physiological Simulation Environment Applying A Hybrid Approach Based On Hierarchical Modeling Using Algebraic And Differential Systems And Dynamic Bayesian Networks

Applicant:
Ciamak Abkai

May the 31st, 2010

Expert Reviewers:
Prof. Dr. Jürgen Hesser
Prof. Dr. Reinhard Männer
Prof. Dr. Colin Atkinson

Ehrenwörtliche Erklärung

Ich versichere, dass die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel von mir angefertigt wurde. Die aus den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen wurden als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mannheim, den 31. Mai 2010

_____

Ciamak Abkai

Danksagung

... I wished to live deliberately,
to front only the essential facts of life,
and see if I could not learn
what it had to teach, and not,
when I came to die,
discover that I had not lived...

Henry David Thoreau (1817-1862)

Dreiandhalb Jahre Dissertationszeitraum, eine Zeit der universitätspolitischen Umwelzungen, in denen ich meiner Zugehörigkeit zu den Universitäten Mannheim, Heidelberg, der Technischen Informatik (TI), dem Institut für Computergestützte Medizin (ICM) oder dem Klinikum Mannheim nicht mehr wirklich sicher war. Obwohl solche Zeiten mit Gewissheit der Forschung nicht dienlich sind, haben Menschen wie **Prof. Dr. Reinhard Männer** und **Prof. Dr. Jürgen Hesser** den Druck und die Unsicherheit abgefangen, so dass Studenten, wie ich uns auf unsere Forschung konzentrieren konnten. Vielen Dank dafür.

Diese Jahre waren zugleich ein Zeitraum, in der ich mit vielen verschiedenen wissenschaftlichen Themen und Ansichten in Berührung kam. Dadurch konnte ich täglich neue Erfahrungen sammeln und meine Perspektive reichlich erweitern, als Mensch und als Wissenschaftler, ganz nach dem Zitat von H.D. Thoreau (s.o.). Die Gesamtheit dieser Erfahrungen ermöglichte eine solche Arbeit, eine interdisziplinäre Arbeit, eine Arbeit, die Software, Hardware, Mathematik, Physik und sogar physiologische und psychologische Aspekte vereinigt. Für diese seltene Gelegenheit bin ich dankbar und auf das Resultat stolz.

An dieser Stelle möchte ich all den Menschen herzlichst danken, die mir dabei - sei es durch Ihr persönliches Engagement, durch technische und wissenschaftliche Diskussionen, Kritiken und Meinungen oder einfach nur durch Aufmunterung - geholfen haben diese Leistung zu erbringen.

- An erster Stelle, danke ich ganz besonders meinem Mentor und Doktorvater **Prof. Dr. rer. nat. Jürgen Hesser**, der mich sowohl stets moralisch unterstüzte als auch jeder Zeit durch sein fachliches Wissen und seinen hilfreichen Ideen in die richtige Richtung lenkte. Natürlich auch für die vielen Reviews unserer Paper und die Optimierungs- und Verbesserungvorschläge.

- Mindestens genauso dankbar bin ich meiner lieben Frau **Anette Abkai**, ohne deren moralische und freundliche Unterstützung über all die Jahre hinweg ich sicher nicht so weit wäre wie jetzt. Sie musste nicht nur die erfolgreichen Tage in der Forschung miterleben, sondern auch die Tage an denen die Praxis anders aussieht als in der Theorie beschrieben.

- Mein besonderer Dank gilt meiner Familie, meinen Eltern **Ilse und Abbas Abkai**, die trotz der weiten Ferne, die sie auf sich luden, damit ihr Sohn sich seiner Ausbildung widmen konnte, stets nahe sind.

- Unermüdlicher Einsatz und Unterstützung wurde mir von meinen Kolleginnen und Kollegen am ICM (Universität Mannheim), sowohl von der Medical Physics Gruppe am Klinikum Mannheim (Universität Heidelberg) und den Mitglieder des ECOMODIS Projektes geboten. Vielen Dank euch allen.

- Mit Frau **Dr. Meike Jipp** habe ich sehr erfolgreich im Projekt ECOMODIS zusammen gearbeitet. Zusammen haben wir einige interdisziplinäre Forschungsbeiträge erbracht.

- Frau **Dr. Ulrike Kornmesser** und Frau **Kathrin Höfer** von der Cathi GmbH danke ich für ihre große Unterstützung rund um alle Fragen bezüglich medizinische Simulatoren.

- **Dr. Nico Becherer**, der mir am Anfang meiner wissenschaftlichen Laufbahn durch seine Unterstützung als Mentor beistand und **Dzmitry Stspankou**, den ich nicht nur als wissenschafler und Kolege, sondern auch als Freund sehr schätze. Beide haben mir gezeigt, dass ein guter Wissenschaftler nicht blind vertrauen darf und sogar kritisch hinterfragen muss.

Contents

1 Preamble

1.1 Introduction

Physiology - as a branch of biology - is generally dealing with functionality of living organisms and its constituents. When Hippocrates 420 B.C. was examining the first human physiological functions like blood pulsation in vessels by observation, probably he was not expecting that one day computer systems would simulate and model this functionality to assist more complex systems e.g. in medical educational simulators or prognosis devices.

As one can see, the history of physiology is more than 2000 years old, and the knowledge of this science has grown constantly ever since. Nowadays, physiology is not only restricted to biology, but also to other sciences including medicine, pharmacology, system theory, engineering, bio-engineering and psychology. This results in a conglomerate concerning, amongst others, metabolism, diseases, organ functions, pharmacological reactions, cell and genome or Deoxyribonucleic Acid (DNA) dysfunctions. Consequently, the short expression, **"human physiological simulation"** can induce a very complex area of science, yielding to ambiguity concerning what is the topic under discussion, especially if scientists from different areas/directions use this terminology.

Although it seems impossible to have a uniform understanding of this expression for different scientific perspectives, there is one common point: every scientist speaking of *physiological simulations*, is using a ***model*** to describe a system - in this case the human physiological system. Therefore system theory and system engineering are increasingly becoming a substantial part of biology, human medicine or psychology. By providing various mathematical methods as well as models, one can describe, analyze and control systems, which may be biological or physiological as well.

While many observations are manifested in simple formulas to describe only parts of human physiology, one is going to the direction of modeling the complete human physiological system by combining several smaller observations into a quite complex overall model. This is the trend of Information and Communication Technology (ICT) in healthcare science over the

past years and is supported e.g. by European and global scientific projects as the Virtual Physiological Human (VPH)[47][108].

The main purpose of physiological modeling and simulation, however, is to serve to healthcare and medical science. The major impact relies on **medical teaching and education**, which are basically dependent on **training medical skills**. Those can be practiced on live and real patients, which allow to develop clinical decision making skills. However, real patients are not available all the time and additionally one cannot always find the right diseases or individual patient for teaching or practicing [117]. Therefore to expose the trainee to sufficient patient scenarios in order to foster mastery of complex knowledge needed, virtual and simulated patients are very useful, which are strongly related to physiological knowledge e.g. in form of physiological models.

Thus, various methods exist to correlate the knowledge about tissue anatomy to the function of the tissue. Of course the process of physiological simulation is not limited to the level of tissues/organs or corresponding vital parameters as Heart Rate (HR). There is a global effort [69][47] started to understand the biology systematically by defining systems and models for different **levels** starting from proteins and genomes to higher levels as blood circulation and regulation. The interesting point is, that the methods for modeling are very similar for all different modeling levels [110], whether micro or macro simulation.

1.2 Motivation

The first physiological simulations had a simple purpose: verifying whether the human understanding (model) of a physiological process conforms to the observation (measurements) of real life cases [97]. Very soon scientists understood that, by the comprehension of the process, one can also describe faulty and malformed physiology and even simulate new approaches to influence and stabilize them [226]. By this, critical situations could be avoided, because the therapy is simulated and tested, before applied to the real patient. Using so-called "off-line simulations" of therapy procedures, clinical applications were optimized strongly during the last years.

Physiological experiments are performed much easier by simulations, in comparison to complex and time consuming animal experiments. Unfortunately the latter are still methods of choice in many universities to demonstrate hemodynamic effects of blood pressure changes of the Cardio Vascular System (CVS) to students and trainees. Although animal experiments can be regarded as an advancement in comparison to practicing directly on human patients, both can

be questioned from ethical point of view. Therefore many educational programs - especially those for beginners - move toward learning by quantitative models and simulations using computational systems [131][122].

Medical simulation environments and especially physiological simulation systems [177] achieve a great increase of interest in the last two decades, by focusing on medical education and training. They are becoming part of todays undergraduate, postgraduate, and continuing medical education curricula [135]. The apparent advantages, in comparison to learning from animal experiments or learning from real patient cases, are :

- **Risk minimization** - There is no risk in making mistakes in a simulation. An incorrect or suboptimal treatment for real patient would perhaps lead to critical or life-threatening situations.

- **Availability** - Difficult patient cases, anomalies, dysfunctions or patient complications are first of all very rare, second too crucial to use them for training of beginners, and third not available or predictable. However, exactly these cases are the important ones. Due to their difficulty level, they need to be practiced and studied to avoid clinical mis-treatments.

- **Repeatability** - By practicing skills again and again trainees learn to manage critical situations in a much better way. Consequently, their treatment skills for complicated patients are experienced and routinely [143].

- **Safety** - It has been shown systematically that simulators (for anesthesia) have improved patient safety importantly. The "human error" was minimized by training crises and emergency management e.g. with full-scale simulators [127].

- **Cost efficiency** - Despite the high costs for purchasing a simulation center, the long-term comparison with instructor payment for trainee education makes the simulation center attractive. Additionally, a more efficient education with simulator systems leads to optimized medical intervention procedures, which also result in a cost efficiency that cannot be ignored [131] [75].

- **Time efficiency** - By improving patient safety, less critical situations occur, the interventions are planned in a better way and therefore the required medical treatment times are also optimized [131].

- **Error management** - In a clinical situation errors must be prevented. In a simulation the trainees can learn from the mistakes, because they have the possibility to understand

the implication of the errors. For crisis management special scenarios can even force the trainees to produce errors, due to overloading, with the purpose to prepare for unexpected critical situations [13].

The variety of medical simulators is very broad and ranges from simple Electrocardiogram (ECG) simulators, over part-scale simulators in cardiology, where physicians can train special skills during interventions, and fully virtual environments for medical process planning to highly complex full-scale simulators. Full-scale simulators deal with various physiological and anatomical models and try to copy real human patients in a very realistic way. They are the pioneers in integrating of both anatomical knowledge and physiological one into a common combined system. A knowledge fusion from different levels of virtual human description [69][47].

In contrast to full-scale simulators, where the subject of education is still related to hardware or augmented reality, fully virtual reality simulators are becoming increasingly popular for medical education since the last years. Applying new methods from Virtual Reality (VR), maybe one day physical mannequin simulators will be fully replaced by virtual ones, e.g. already used for virtual anesthesia [85], surgery simulation [25] and emergency planning in second life [98].

The role of medical and physiological simulations is however not limited to academic education or training, this topic is highly actual in medical research and engineering for medical improvements and thus also from industrial point of view or military (e.g. medic training) [57].

Although still far away to treat diseases fully by computational systems, nevertheless physiological simulations and computations already play a very important role in robot-guided therapy and intervention [57] and are also becoming popular in other medical systems [8], e.g. functional imaging for a better diagnostic and therapeutic effort.

Hence, physiological simulations are an *actual field of research* and attract worldwide research attention. The reason is simple, by understanding the physiological process and establishing connections to other scientific domains, one is able to provide better methods in health care. One good example is the development of physiological based pharmacological models, which provide a better understanding of drug delivery and biochemical processes based on physiological models and thus support the clinical personal in more effective way e.g. by patient individual drug dose planning [191].

Another very important and relatively new field of integration is functional imaging, because

more and more physicians are not only interested in human anatomy, but also in the physiology behind. A good example here is the *J4 Imaging System*, which is a quantitative medical imaging software toolbox, focusing primarily on PET, MRI and CT applications for cardiology, allowing integration of physiological knowledge by modeling and simulation [69].

1.3 Purpose Of The Work

As it was mentioned in the section *Motivation* and will be pointed out more clearly and in detail in chapter 3 *State of the art* (page 49), current physiological simulation systems have many shortcomings. Even commercial teaching simulators - whether full or part-scale - could greatly benefit from better models.

Known problems and limitations of educational simulator systems are the following:

- Medical part- or full-scale simulators are very cost expensive ($ 600.000 for a simulation center [86]).
- Many different simulator systems have no (or no open) standard. Therefore it is very difficult to integrate different systems or different system components together.
- Many simulator systems do not support physiological simulations at all. Other do only use a storage of pre-recorded patient data [133]. Physiological dynamics - by the meaning of reaction to different interventions - are partially very unrealistic or there is no dynamic reaction at all [194].
- Physiological computing simulations are only established in one (known state of the art at Dec. 2009) full-scale simulator (HPS from Meti [159]), however, computations do not always match to expert knowledge or results from literature [194].
- While full-scale simulators allow to access patient vital parameter as realistic as possible by simulation and emulation (e.g. acoustic sound of heart, pupil reactions), many simulation systems does this only virtually (screen based). Unfortunately there is no standard or systemic way to fix the incompatibility of different simulation hardware yet.

Known problems for physiological simulation and modeling software are:

- Physiological simulation and modeling software are existing, but they are not standardized yet. It's very difficult to transfer models from one environment to others.

- Many software environments (e.g. GasMan [78], BodySim [26]) do not allow to change or extend models. Even the well composed system SAPHIR [223], which includes the Guyton's circulatory system [88] is hard coded into the software and can only be manipulated by experienced programmers. They are inflexible in their design.

- Although known systems, e.g. JSIM [182], QCP [186], SBML [202], try to overcome the above mentioned problems, by providing description languages (Mathematical Modeling Language (MML), Extensible Markup Language (XML), Systems Biology Markup Language (SBML)), which could be used as a standard for physiological modeling, there are still major problems, like non-hierarchical modeling and thus a huge design complexity.

- Due to the complexity of physiological models, hierarchical modeling is needed, which is not supported by Physiome or QCP. This problem is also well known from system biology modeling [104][202].

- The existing compiling tools (Physiome [69] [52] and QCP [9] [186]) do not consider real-time capability. They only provide off-line simulations.

- Many physiological dependencies are - due to the lack of knowledge - based on stochastic methods (e.g. population modeling). Many causal relations are afflicted with uncertainty. Unfortunately, the mentioned tools do not support non-deterministic mathematical approaches yet.

Therefore this work focuses on the problems, current systems are faced with, and look exactly to those regions, where working systems are showing weaknesses and possibilities for potential improvements. This includes in parallel the integration of new modeling approaches for dynamic systems. These goals can be summarized as following:

- Provide a modeling language and system, which support both: first, hierarchical modeling and second hybrid deterministic and probabilistic systems. The proposed system will support systems of Ordinary Differential Equation (ODE) and static and dynamic Bayesian Network (BN). The language will be XML oriented, such that it can provide further extensions and changes managed by a version control system.

- Show the benefits of the suggested modeling approaches, especially in cooperated design of complex physiological systems, with huge amount of parameter uncertainty.

- Provide a compiling and processing tool for the suggested modeling language, which is real-time capable and can thus be used for integration into real time medical educational simulators.

- Provide a client server based methodology, which can be used as a standard for integration of hardware emulation devices into medical educational systems, whether part- or full-scale simulators.

- Proof the concept of the client server based methods by providing three by the state of the art not existing emulation systems, important as well for medical education as for medical equipment reliability testing.

- Deliver an ECG simulation hardware interface based on multiple dipoles for 12-channel ECG and full control of electrophysiological models, including dynamic attributes.

- Deliver a SaO2 simulation hardware interface, which can be attached to any monitoring system and includes full control of the emulation parameters (pulse, oxygen saturation).

- Deliver a IBP simulation hardware interface, capable so emulate pressure dynamics, accessible by any catheter pressure sensor system and including full control over emulation parameters (pulse, systolic and diastolic pressure curves).

- Proof the overall concept of the suggested methods and models by overall integration of them into a virtual human patient simulator, including emulation (ECG, SaO2, IBP) and simulation (Circulatory system, CVS, respiratory system, cardiovascular medication).

- Examine how far BN and Dynamic Bayesian Network (DBN) mixtures can be used for medical simulations.

- Consider novel approaches of dependability analysis, containing real-time physiological simulations for patient monitoring systems.

By this thesis and the underlying developments, we have proved, that it is possible to build up an open-source environment for medical physiological simulations, including the integration of hardware emulation devices. The new system has several advantages in comparison to existing commercial systems e.g. the full-scale simulator HPS. Just to mention reduced costs, flexibility to extend, manipulate and exchange models.

1.4 Requirements

The core of this thesis will therefore be separated in three parts:

In the first part a general system framework is described allowing real-time capable physiological simulations. This framework should be flexible and extensible for future extensions and be general to allow simple task-specific simulations (e.g. clearance of a specific drug) on the one hand, and on the other hand support a fully and complex physiological simulation as well toward emulation of this signals as done in full-scale simulators. The environment should not only support current state of the art physiological modeling techniques, but also extend the capabilities by novel considerations and methods as the following:

- To address modeling complexity issues, a new object-oriented method will be provided, which allow hierarchical modeling techniques, which were not supported yet in this way.

- To address stochastic modeling approaches and the huge amount of uncertainty in clinical data relations, a new approach is introduced by integration of BNs and DBNs to model probabilistic causality. Also the combination of both - deterministic and probabilistic - methods seems to be very promising. In todays praxis usually linear models are used to fit clinical data, however this approach leads to problems, if the system is non-linear and the observed data sparse and uncertain. By our hybrid approach, we can overcome this problems.

- A language (based on XML) and system for processing of the suggested models and underlying methods will be presented as well.

In the second part there will be a technique suggested, how generally medical education and training systems with emulation of human patient parameters - as usually used in many full-scale simulators - can be integrated by a common standard, based on a client server approach. To proof this concept, three important vital parameter emulation devices (not supported yet by the state of the art) have been developed during this thesis and are integrated into the general framework.

In the third part the suggested technology will be examined for modeling in medical domain, focusing on psychological and physiological modeling and reasoning. Theoretical and practical aspects related to medical system modeling are considered herewith. Also the very relevant relation toward medical patient monitoring is touched within this context and we provide new and interesting theoretical and practical knowledge for this field.

The first and second part match perfectly together, because they show the realization by hardware and software co-design. The result of the overall methods is a human patient simulator, which is based on physiological modeling and supports important vital parameter emulation. This simulator can be used for medical educational programs as well as for patient monitoring

system reliability analysis and testing. In the third part, the importance of the system and the suggested underlying novel methods for modeling, reasoning and dependability analysis in medical domain is examined.

The requirements to the integrative software environment are very strict. First of all, it has to support a dynamic model compiling framework, which is capable to load and unload models (if necessary during the runtime). Additionally to provide a real-time capable system - usually vital parameter needs to be updated with 1-10 kHz frame rates - the computational kernels of the software must be very efficient and optimized.

Additionally, the suggested system should support a good access for object manipulation and visualization interface, which is realized by the Model View Control (MVC) design pattern. Multi threading, unit-testing, version-control, continuous integration, optimization by profiling and log-management - just to mention some - are other important approaches which are applied to provide a common and useful software environment.

1.5 Formal Comments

Citations from literature are marked by "[...]" and long full citations with "..."; the full list of referenced literature is presented at the end of this work. For this thesis a very intensive literature research was applied, continuously updated during the work. In addition to books, reports, papers and journals, technical reports, patents and websites are used to provide a full overview of the state of the art.

Materials, drafts or pictures, which has been used from other authors are published by previously admitted permissions[1] according to actual state of law[2].

To have a better overview of the thesis, a list of figures, a list of tables and a list of acronyms are supported. In the topic under consideration many abbreviations or usually also used in the literature. To avoid confusion, a list of abbreviations is included as well. The formating, layout as well as content guidelines are applied according to German national library[3].

[1] Permissions are admitted mostly directly or by http://www.copyright.com
[2] (§ 51 UrhG) see also http://www.dissonline.de/recht/
[3] http://www.dissonline.de/: Empfehlungen zur Erstellung digitaler Dissertationen mit Latex

1.6 Project Related Publications

Various papers have been published by the author during this work to demonstrate the scientific importance of the theme and show partial results. First of all there are two publications on design of specific (ultrasound) simulators for clinical education [4][31], which are however relevant, because from the system design point of view many components are identical, especially if one thinks of software constructs to assure real-time synchronizations.

Additionally, there are two publications in the field of cognitive modeling, where methods from Bayesian Networks theory have been applied [120]. One will see during the next chapters of this thesis, that cognitive modeling is strongly connected and also very important for many physiological reactions, however only few groups consider this fact in their modeling approaches. In [121] the simulation environment, was used to estimate psychomotoric parameters, which have been learned before by a BN model.

The environment for medical simulation was presented in [6] as well as some emulation interfaces and specific modeling approaches for multiple dipoles to model and simulate vital ECG signals [5].

Finally, end of 2009 methods were introduced to apply physiological models for a new approach in physiological monitoring [8] [7].

1.7 Outline

First, in chapter *Basics* the theoretical background concerning this work will be discussed and shortly explained, such that the interested reader, who is not familiar with the topic, can get a fundamental understanding of the problems and methods. This will mainly include actual models for physiological simulations based on ODEs and some additional explanation about stochastic models and approaches, especially BNs. The explanations are based on current state of the art methods, never the less in the following chapter *State of the art* there will be a highly compressed overview of scientific methods, models and systems which are connected to the topic under discussion.

After these introductory chapters, where the advantages and disadvantages of different methods and systems, are carefully pointed out, in chapter *System Architecture*, there will be a theoretical explanation of the novel methods and ideas, integrated into this project. Optimizations consider the modeling level, software and hardware architecture.

Next, in chapter *Realization*, one can find detailed information about the implementation strategies of the theoretical parts mentioned in the chapter before.

Due to the importance of the emulation hardware interfaces for medical simulators and to proof the concepts of generic hardware architecture for medical signal emulation, a standalone chapter is dedicated to *Vital Signal Modeling and Emulation*, including vital function modeling, hardware emulation, system design, characteristics and results for three different medical patient parameter emulation devices, IBP, SaO2, and ECG. In these chapters, in addition to hardware implementation details, underlying novel mathematical models are provided as well.

Next, in chapter 7 *Mixtures of BN/DBN for Medical Simulations*, novel theoretical modeling approaches, based on dynamic and static BNs, are discussed and practical results are presented. The focus relies hereby on physiological and psychological simulations, which - as will be shown - both are very important for clinical and medical problems.

While in the previous chapters different components concerning spatial parts of the proposed system are discussed in detail, e.g. by focusing on modeling approaches, software or hardware architectures and components, in chapter 8 specific simulations (case studies) are presented, which partially integrate many different components into overall and complex systems. Such models are containing haemodynamic simulations, blood regulations, medications, heart disease influence on electro-physiology and many others.

After showing the results of different modeling and simulation scenarios, run on the system, in chapter 9 *Towards Dependable Monitoring* (page 159), there will be an outlook towards novel approaches using physiological simulations for intelligent especially dependable patient monitoring. As will be shown, dependability analysis of physiological systems is strongly related to simulation and modeling of dynamic system states.

Finally, in the chapters *Discussion*, *Summary and Outlook*, the overall results and underlying models and methods, presented during the work, will be summarized. Herewith possible deficiencies of the presented approaches are pointed out and will show the direction of future research.

2 Basics

In this chapter, essential modeling knowledge will be presented in very summarized form. First compartments and electrical networks and the corresponding solutions of such systems will be discussed. Then probabilistic approaches with static and dynamic BN will be introduced briefly.

2.1 Compartment Modeling

Compartments are well-known in the filed of system theory and system modeling, and are normally used to mathematically describe how materials or energies are propagated through a system. A system can herein consist of one or many compartments, while each compartment is assumed to be homogeneous [1].

In physiological modeling, compartment models have been widely used, due to the easy understanding of different organs or sections of the body (assumed as homogeneous) represented by homogeneous compartments, which are connected the same way as organs. Also in Pharmacokinetic (PK) and Pharmacodynamic (PD) modeling, compartments are used to represent drug concentrations in different sections of the body or different phases of metabolism [221]. The science of PKs is considering the distribution, absorption and elimination of drugs and is therefore very related to physiological modeling, which can lead to the quantification of these processes, leading to the understanding, interpretation, and prediction of blood concentration-time profiles [3].

2.1.1 1- and 2-Compartment Models

The simplest compartment model is the 1-compartment model also known as a 1-tank system, which is depicted in figure 2.1 on left side. As one can see, the change of the volume $V(t)$ over time is described by the following formula, while $I(V)$ is the inflow into the tank and $O(V)$ the outflow from the tank, both assumed to be functions of the volume in the tank:

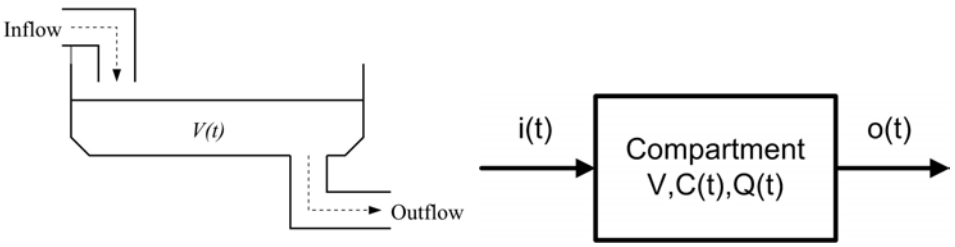


Figure 2.1: 1-compartment model, represented by 1-tank system (left figure) or a symbolic system (right figure). The compartment itself can represent a volume (V), a concentration (C), a mass (Q) or other physical and non-physical properties. Mass, volume and concentration are easily to transform to each other by the relation $Q(t) = V \cdot C(t)$.

$$\frac{dV(t)}{dt} = I(V) - O(V) \tag{2.1}$$

In pharmacology one is usually interested in the mass or concentration of the drug inside the system, in biology in a single-cell concentration monitoring or for radioactivity modeling in the number of radioactive[1] atoms present [1]. All these different views can be modeled by compartments.

Therefore, in general, the dynamic behavior of a value (representing mass, concentration, volume and another physical or nonphysical property) $X(t)$ described by a compartment model, is given by the following linear relation:

$$\frac{dX}{dt} = K \cdot X \tag{2.2}$$

The parameter K is describing the linear relation, e.g. for negative K the decrease and for positive one the increase of $X(t)$. Of course this is related to the input and output and usually we assume $I(X) = k_i \cdot X$, $O(X) = k_o \cdot X$ and $K = k_i - k_o$, which explains easily the relation between the equations 2.1 and 2.2.

This simple relation is describing a **first-order linear differential equation**, which solution is given by integration as following:

[1] The first compartment models were introduced by Rutherford et al. for the description of radioactive decay and thus the related equations are known as Rutherford equations. [1]

$$\int_{t_0}^{t} \frac{X'}{X} dt = \int_{t_0}^{t} K dt \tag{2.3}$$

substitution $U = X(t)$ lead to:

$$\begin{aligned} \int_{X(t_0)}^{X(t)} \frac{1}{U} du &= \int_{t_0}^{t} K dt \\ ln(\frac{X(t)}{X(t_0)}) &= K(t - t_0) \\ X(t) &= X(t_0) \cdot e^{-K(t-t_0)} \end{aligned} \tag{2.4}$$

while $X(t_0)$ is the value of $X(t)$ at time $t = t_0$, which is defining the initial conditions of the system. Equation 2.4 is describing one solution of the differential equation, which can be extended by constant additive parameters. This underlines the importance of the initial conditions for a unique solution.

In a pharmacokinetic model, the compartments may represent different sections of a body [3], where the concentration of a drug is assumed to be uniformly equal[2], as depicted in figure 2.2.

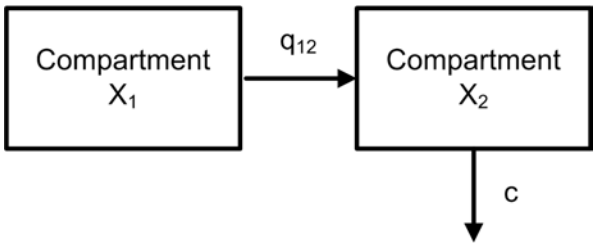


Figure 2.2: 2-compartment model with inter-compartmental clearance q_{12} from compartment 1 to compartment 2 and total system clearance c from compartment 2.

Then the model is described by a system of linear differential equations as given by equation 2.5 for two compartments 1 and 2.

[2] An underlying assumption of compartmental models is that the material in the compartment is instantaneously well mixed [1].

$$\begin{aligned} \frac{dX_1}{dt} &= -q_{12} \cdot X_1 \\ \frac{dX_2}{dt} &= +q_{12} \cdot X_1 - c \cdot X_2 \end{aligned} \tag{2.5}$$

Here, one can apply integration from equation 2.4 to describe the overall drug concentration $X_2(t)$ by a biexponential representation [1]:

$$\begin{aligned} X_1(t) &= X_1(t_0) \cdot e^{-q_{12}(t-t_0)} \\ X_2(t) &= \frac{q_{12}}{c - q_{12}} \cdot X_1(t_0) \cdot e^{-q_{12}(t-t_0)} + (X_2(t_0) + \frac{q_{12}}{q_{12} - c} X_1(t_0)) \cdot e^{-c(t-t_0)} \end{aligned} \tag{2.6}$$

Comparing to figure 2.2 the parameter q_{12} describes the inter-compartmental clearance from compartment 1 to 2 and c is the clearance [192] from compartment 1.

2.1.2 N-Compartment Models

Although the 1- and 2-compartment (see figures 2.1 and 2.2) models may seem to be too simple to describe complex systems, they essentially represent the basic ideas and the method behind modeling using compartments. Considering the 2-compartment model given by figure 2.2, the resulting analytical equations to describe the system behavior are already quite complex. Nevertheless, just by combining the two presented models to higher order N-compartment models, one can describe complex system behavior attributes over time. The fundament of the description, however, is kept very simple and easy to understand.

Motivated by Rutherford's equations for successive transformations, multi compartment systems are widely used to describe first order processes. Actually in PK and PD modeling usually up to nine particular compartments are used to achieve reasonable solutions. In general, such a system of N linear differential equations can be described by the following equation.

$$\frac{dX}{dt} = K \cdot X + u(t) \tag{2.7}$$

while X is the state of the system and K is a matrix describing the coefficients. u is the vector for independent changes in case of a non-homogeneous system of ODE. For a N-dimensional

ODE system, K could be described by a matrix (including only constant parameter K_{ij}, $i, j = 1..N$), and the system would look like the following:

$$\begin{pmatrix} dX_1/dt \\ dX_2/dt \\ \vdots \\ dX_N/dt \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & \cdots & K_{1N} \\ K_{21} & K_{22} & \cdots & K_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ K_{N1} & K_{N2} & \cdots & K_{NN} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{pmatrix} + \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_N(t) \end{pmatrix} \tag{2.8}$$

As one has seen from equation 2.6, already an exact solution for a system with two compartments is leading to complex analytical expressions and could only be calculated under the assumptions that the (inter-)compartmental clearances are constant and independent. The general solution for equation 2.8 is given by applying an eigenvalue decomposition, leading to:

$$X = \sum_{i=1}^{N} c_i S_i e^{\lambda_i t} + \sum_{i=1}^{N} e^{\lambda_i t} S_i \int e^{-\lambda_i t} S_i^{-1} u(t) \tag{2.9}$$

while $K \cdot X = \lambda X$ is given by an eigenvalue decomposition with eigenvalues λ_i and eigenvectors S_i, with $S = [S_1, S_2, ..., S_N]$, $i = 1..N$ and c_i describe zero time conditions.

As one can see, the analytical solution of such a system is firstly dependent on an integration process with the function $u_i(t)$ involved and secondary includes despite the integration many exponential sums.

To overcome these complexity issues and restrictions, usually the solutions for such systems are not calculated analytically but by numerical integration methods, which will be discussed in section 2.3. Before going to details about differential equation numerical integration strategies, there will be a short comparison with another modeling method - closely related to compartment modeling - by using electrical equivalent circuits in the next section.

2.2 Electrical Equivalence Of Physiological Systems

Guyton's circulatory modeling system [88] describes blood flow in the circulatory system by concerning many dynamic attributes (e.g. Windkessel effect in arterial system [130]). By this, a systematical consideration and modeling of the CVS can support clinical interpretations and

understanding of cardiovascular diseases [218].

The blood pressure and flow waveform changes in shape and amplitude between heart and more distal parts of the vascular structure. The changes occur due to interactions between elastic arterial wall (capacitance effect), blood mass (inertial effect), and frictional (resistive) effects. In the more distal arteries and arterioles frictional (resistive) effects on blood flow become increasingly important.

However, knowledge about dynamics of circulatory system is sufficient to fill easily several books [218], in the following subsection a selection of few relevant principles is presented, describing the above mentioned interactions in the circulation system briefly.

2.2.1 Resistive Element

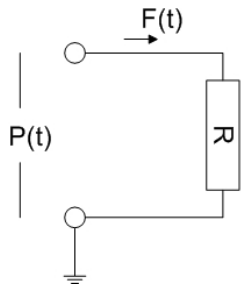


Figure 2.3: Resistive element: this model is used to describe friction and resistive attributes to pressure in dependency to flow throw a vessel or segment, given by the Ohm's law.

The resistive element (see figure 2.3) of a circulatory system can be expressed in terms of an electrical analog. The voltage V - which is a potential difference - is represented by the pressure difference P. Although SI units[3] are commonly used in scientific physical considerations, for physiological and medical statements typically units and dimensions are used, that are closely related to the history, how physical parameters are measured in clinical daily routine. Thus pressure information are usually provided with dimension millimeter of mercury (mmHg).

The current I represents the flow-rate F, which is typically measured in milliliter per second (ml/s) and the electrical resistance is corresponding to a vessel resistance R, which describes how much the pressure will change in relation to the flow-rate (dimension $mmHg \cdot s/ml$). According to the Ohm's law the following formula for the circulatory resistive element can

[3]International system of units (abbreviated SI from the French "le Système international d'unités". Although in the following and in literature usually clinical units (e.g. $mmHg$ instead of Pa) are used, they are equivalent and can be easily transformed to each other (e.g. $1Pa \approx 7.5006 \cdot 10^{-3} mmHg$).

therefore be used:

$$P(t) = R \cdot F(t) \tag{2.10}$$

The resistive element is thus describing a rigid vessel by a friction term, resulting into a pressure drop. The vessel resistance R can be determined by the Hagen-Poiseuille relation, using the following equation, while r is the radius of the vessel, l the length of the vessel and μ is the fluid viscosity:

$$R = \frac{8\mu l}{\pi r^4} \tag{2.11}$$

Of course, this formula is based on a physical simplification, considering the temperature and the fluid dynamics to be constant. For blood, usually a viscosity of $\mu \in [4 - 25]\ mPa \cdot s$ is assumed (for an average temperature of 37 °C). For a vessel with radius $r = 0.03\ cm$, and a length $l = 1\ cm$ this will result in a resistance $R \in [94 - 590]\ mmHg \cdot s/ml$. Equation 2.11 clarifies that the most pressure drop will be in the capillaries, where the diameter of vessels is about 5 μm and the length about 0.5 mm, resulting in a comparable big resistance $R \approx 382 \cdot 10^6\ mmHg \cdot s/ml$ for one capillary.

2.2.2 Elasticity / Compliance

In figure 2.4 the basic model for the compliance of vessels in circulatory systems is depicted. It simulates the current flow $F(t)$, into an elastic vessel with compliance C. The analogous description in terms of fluid flow in a blocked vessel is that an initial pressure $P(t)$, applied across the vessel, causes an increase or decrease in vessel volume V, accompanied by a flow change F, into or out of the vessel.

Again electrical equivalences can be used. The fluid analogous for capacitance is the compliance C (with dimensions $ml/mmHg$). The changes in volume $\frac{dV(t)}{dt}$ can be described as following:

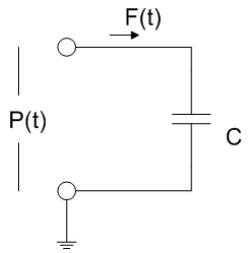


Figure 2.4: Compliance element: By this model, the effect of stiffness or elasticity of vessels to the time varying vessel flow $F(t)$, depending on the input pressure $P(t)$ is described by a first order differential equation.

$$\frac{dV(t)}{dt} = \frac{dP(t)}{dt} \cdot C \tag{2.12}$$

$$F(t) = \frac{dV(t)}{dt} \tag{2.13}$$

$$F(t) = \frac{dP(t)}{dt} \cdot C \tag{2.14}$$

As one can see, this results in a formulation for the flow, depending on variations of pressure $\frac{dP(t)}{dt}$.

The compliance parameters C of a basic elastic vessel can be approximated according the law of LaPlace (assuming isotropy) and the Pascal's principle [243][182] as the following:

$$C = \frac{2D_0^3\sigma^2E^2\pi l}{2\sigma E - PD_0} \tag{2.15}$$

Hereby D_0 is the diameter of the vessel at zero stress, σ is the thickness of the vessel wall, l is the length of the vessel and E is the Young's modulus [243], defined by $E = \frac{PD}{2\sigma}\frac{D_0}{D-D_0}$ for D as the diameter under stress and P the pressure under stress.

Typical model values can be found in table 2.1:

Table 2.1: Typical values for vessel compliance

Variable	Value	Meaning
σ	20 μm	Vessel wall thickness
l	12.5 mm	Vessel length
D	200 μm	Vessel diameter without stress
D_0	150 μm	Vessel diameter under stress
E	450 $mmHg$	Young's modulus for vessel wall
P	30 $mmHg$	Driven pressure under stress
C	0.0087 $\mu L/mmHg$	Resulting compliance

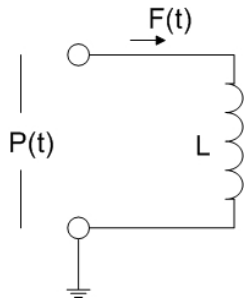


Figure 2.5: Inertance element: By this model the effect of inertance in vessels is described, which is a form of energy storage. The relation is given by a first order differential equation.

2.2.3 Inertance

Inertance is a measure of the pressure gradient in a fluid required to cause a change in flow-rate with time. The electrical analogue is given by an inductance (inductivity), as depicted in figure 2.5.

The time varying pressure $P(t)$ and flow-rate $F(t)$ relation to the inertance L (dimension = $\frac{mmHg \cdot s^2}{ml}$) is hereby given by the differential equation 2.16.

$$\frac{dF(t)}{dt} = \frac{P(t)}{L} \tag{2.16}$$

The arterial inertance L for a vessel segment with length l and cross-sectional area A can be approximated by the following equation [215]:

$$L = \int_0^l \frac{\rho \cdot c_v}{A(x)} dx \tag{2.17}$$

while ρ is the blood density, $c_v \approx 4/3$ is a non-flat velocity coefficient. A typical value for total arterial inertance validated with real date is $L = 0.0054\ mmHg \cdot s^2/ml$ [215].

2.2.4 Elastic Vessel

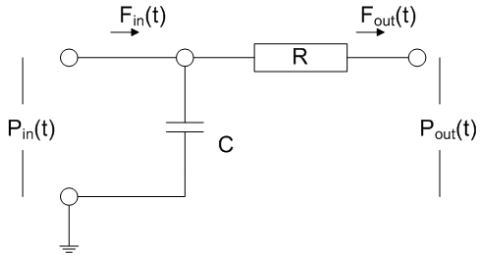


Figure 2.6: Elastic vessel: A 2-element vessel model for modeling of dynamics of vessels. It is based on a resistive element and a compliance element, resulting in an electrical RC single pole filter.

Usually one is interested in the modeling of the dynamic attributes of vessels, that are essentially elasticity and energy storage. Therefore, in praxis, basic models are usually combined to more complex fluid circuits. The model, depicted in figure 2.6, simulates the flow, volume and pressure of a compliance, resistive vessel - the simplest formula for a dynamic vessel model (a 2-element model).

An input pressure P_{in} drives the flow F_{in} in the vessel. The flow into the vessel F_{in}, the flow out from the vessel F_{out} and the flow due to compliance F_{comp} are unknown parameters in this simulation. Therefore the following three equations are necessary, which were partially introduced before.

$$F_{out} = \frac{P_{in} - P_{out}}{R} \tag{2.18}$$

$$F_{in} = F_{out} - F_{comp} \tag{2.19}$$

$$F_{comp} = \frac{dV(t)}{dt} = \frac{dP(t)}{dt} \cdot C \tag{2.20}$$

The first equation describes the flow out from the vessel and is related to the resistance by the fluid equivalent of Ohm's Law resulting from equation 2.10.The flow into the vessel and the flow out of the vessel (second equation) are different, because of the change in volume. The last equation describes this changes, resulting from equation 2.12.

As one can see, by this the elastic attributes of vessels are modeled and the reaction to pressure changes at the input of a vessel are essentially described by a low-pass filter behavior. The 3-element model, as depicted in figure 2.7, is an improvement of the model before and is already providing a good approximation of the vessel dynamic behavior [215]. However, to provide a better fit to the so called windkessel effect, at least a 4-element model is usually applied, as will be presented in the next subsection.

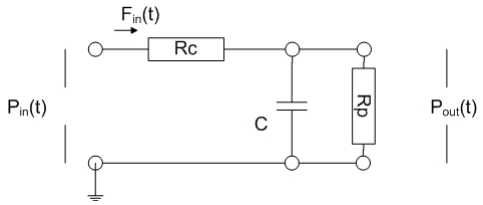


Figure 2.7: Elastic vessel: A 3-element vessel model for modeling of dynamics of vessels, including characteristic vessel resistance R_c, peripheral resistance R_p and compliance C.

2.2.5 Windkessel Effect / Vessel

The windkessel is a description for elastic reservoirs [91]. Especially for large arteries (so called windkessel vessels) a change in vessel diameter is observable (so called Windkessel effect). This is because the walls of large arteries (eg. aorta, common carotid, subclavian, and pulmonary arteries and their larger branches) contain elastic fibers. The change of vessel diameter during systole and diastole results in a bigger volume of the vessel during systole phase, which is acting on the one hand as a low-pass filter and on the other hand as an energy puffer.

One method to describe the mentioned effects, is based on the 4-element windkessel model [215], which is given by figure 2.8. This model is similar to the elastic vessel model from the previous chapter and additionally includes an inertance element.

This system is fully described by the following equations (5 algebraic and 2 ODE):

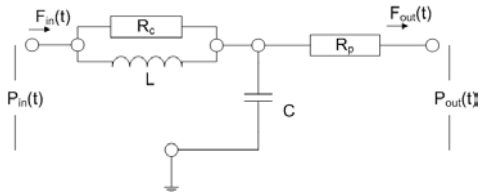


Figure 2.8: Elastic vessel: A 4-element Windkessel model for modeling of dynamics of a vessel, including the characteristic vessel resistance R_c, the peripheral resistance R_p, the inductance L and the compliance C.

$$F_{in} = F_{Rc} + F_L \tag{2.21}$$

$$F_C = F_{Rc} + F_L - F_{out} \tag{2.22}$$

$$P_{in} = P_{Rc} + P_p = (F_{Rc} \cdot R_c) + P_p \tag{2.23}$$

$$P_p = \frac{V_p}{C} \tag{2.24}$$

$$F_{Rp} = \frac{P_p - P_{out}}{R_p} \tag{2.25}$$

$$\frac{dF_L}{dt} = \frac{P_{in} - P_p}{L} \tag{2.26}$$

$$\frac{dV_p}{dt} = F_C \tag{2.27}$$

R_c stays here fore the characteristic resistance of the vessel, while R_p is representing the peripheral resistance. As one can see, the first 3 equations are resulting from the Kirchhoff's circuit laws and the other equations are just representing the knowledge from equations 2.10, 2.12 and 2.16.

2.2.6 Vessel / Circulatory Systems

Normally, in hemodynamic modeling, one is interested in networks and larger systems containing several vessels. This is achieved by the combination of smaller vessel representations and the integration into a larger system, as depicted in figure 2.9 [226]. Here a representation for arterial flow is given, which includes the resistance R_{ao} and the inertance L_{ao} for the aorta, the resistance R_{ea} and the compliance C_{ea} representing epicardial arteries, and the same representation for large arteries (la), small arteries (sa) and capillaries (cap). The common ground pressure is the pressure in the interstitial fluid P_{isf}.

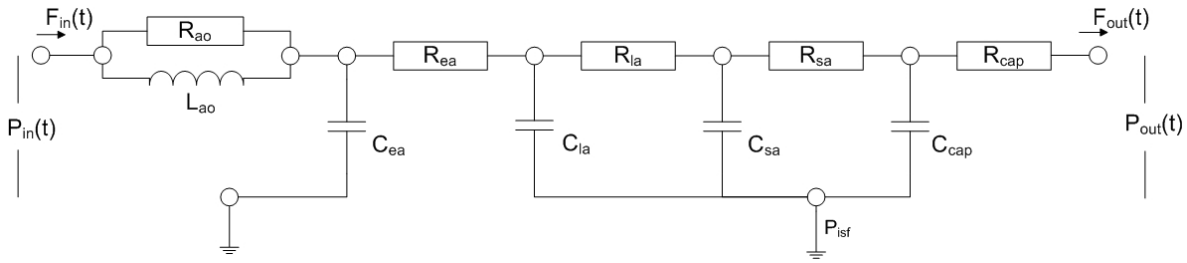


Figure 2.9: Vessel system: An example of arterial system is depicted, including resistances R, inductance L and compliance elements C. The following segments have been represented: aorta (ao), epicardial arteries(ea), large (la) and small arteries (sa) and capillaries(cap).

By this, larger circulatory systems e.g. the coronary, pulmonary and systemic circulation can be constructed. They provide the basics for haemodynamic modeling of an highly integrative human model [182][91]. If the flow is known for larger vessels and thus for organs, one can use the information to calculate the distribution of substances, e.g. drugs inside the circulatory and organ system (see chapter 3 - "State Of The Art").

As previously indicated, compartment models and fluid circuit models (described in this section) are closely related. Not only due to the dependency between flow (fluid circuit) and drug distribution (compartment), but also due to the nature of both descriptions, which are based on systems of ODEs.

Such differential systems can be analyzed and threated in various ways (e.g. analytically, numerically, in frequency and time domain) with different outcome. In this work, we are interested in real-time simulation (mimicking) of the physiological behavior of a real patient and therefore in sampling of solutions of the mentioned systems. One way to fulfill this task, is to apply numerical ODE integration techniques, which will be discussed in the next section.

2.3 ODEs And Numerical Integration

As one can see, the previously presented modeling approaches (compartment models, electrical equivalent modeling) are based on systems of ODEs [154]. In this section, there will be an introduction to ODE problems and solutions. The general single initial-value ODE problem can be described by the equations 2.28.

$$\frac{dy}{dt} = f(y,t), \qquad y(t_0) = y_0 \tag{2.28}$$

Here, y is the dependent variable, and t is the independent variable, $f(y,t)$ is a derivative function, t_0 is the initial value of the independent variable and y_0 is the corresponding initial value of the dependent variable. If only first order differential equations are considered, which is typically for LTI systems and compartment models, the derivative function can be represented by a Jacobian Matrix [140].

2.3.1 Taylor Expansion

In contrast to analytical solutions, which could exist for the above problem, **numerical solutions** are applied to compute pairs (t_i, y_i), fulfilling the criteria given by equation 2.28, starting with the initial condition pair (t_0, y_0). The process of numerical integration is defined by a step by step algorithm calculating a new pair (t_{i+1}, y_{i+r}) based on a previous pair (t_i, y_i). This stepping procedure can usually be determined by a Taylor series, as shown in equation 2.29.

$$\begin{aligned} y_{i+1} &= \sum_{n=o}^{\infty} \frac{h^n}{n!} \frac{d^n y_i}{dt^n} \\ y_{i+1} &= y_i + \frac{dy_i}{dt} h + \frac{d^2 y_i}{dt^2} \frac{h^2}{2} + ... \end{aligned} \tag{2.29}$$

In praxis, a finite number of summation terms are used for approximation and the Taylor series is truncated. $h = t_{i+1} - t_i$ is determined by the step size of the independent variable.

By only applying the formula up to a linear term of h, already a good approximation is given by the eq. 2.30.

$$y_{i+1} \approx y_i + \frac{dy_i}{dt} h = y_i + h \cdot f(y_i, t_i) \tag{2.30}$$

This is the explicit *Euler's method* (forward Euler method), which is a linear approximation

and only requires the first derivative $\frac{dy_i}{dt}$ for $t_i = t_0 + i \cdot h$. The smaller the step size h is chosen, the better is the approximation. Nevertheless, there will be a small error (truncation error), in comparison to an exact analytical solution, because the Taylor series was truncated. This error is step size sensitive and thus many methods focus to control the error by varying the step-size h (adaptive integration methods).

In contrary to the explicit method, in the implicit Euler's (backward Euler) method , the solution is give by:

$$y_{i+1} = y_i + h \cdot f(y_{i+1}, t_{i+1}) \tag{2.31}$$

Here, the value of $f(y_{i+1}, t_{i+1})$ is only implicit known and thus one has to solve a linear or non-linear equation to find the solution of y_{i+1}.

2.3.2 Higher Order Methods

Obviously, the truncation error - as mentioned before - will be smaller, if higher order terms of the Taylor series (eq. 2.29) are considered as well. For this purpose, the function derivatives must be known and calculated for each step. Especially for real-time application, complex derivative functions could lead to a bottleneck regarding computational time. Therefore, alternatively approximative strategies can be applied, e.g. **finite difference approximation** [92].

This approach is used in the so called **Runge-Kutta methods**, where intermediate states are calculated in each iteration (see second order Runge-Kutta known as the Heun's method) [140][92]. The Runge-Kutta fourth order (RK4-method) is using the approximation of order of four and a single step method with step-size h, as given by equations 2.32.

$$
\begin{aligned}
y(t_0) &= y_0 && ,\dot{y}_0 && = f(y_0, t_0) \\
y_A &= y_0 + \frac{h}{2} \cdot \dot{y}_0 && ,\dot{y}_A && = f(x_0 + \frac{h}{2}, y_A) \\
y_B &= y_0 + \frac{h}{2} \cdot \dot{y}_A && ,\dot{y}_B && = f(x_0 + \frac{h}{2}, y_B) \\
y_C &= y_0 + h \cdot \dot{y}_B && ,\dot{y}_C && = f(x_0 + h, y_C) \\
y_1 &= y_0 + h \cdot \frac{1}{6}(\dot{y}_0 + 2(\dot{y}_A + \dot{y}_B) + \dot{y}_C) && && (2.32)
\end{aligned}
$$

Other solvers, using higher order derivatives, are **Hermite Obreschko** and **Fehlberg** methods (O(h^5)-method) [111]. The **Parker Sochacki** method computes the coefficients of the Taylor series recursively [140]. The interested reader can find additional information about numerical methods for solving of differential equations in [92]. Furthermore, different optimized methods are offered by the GNU scientific library (GSL)[87].

2.4 Bayesian Networks

Whenever reasoning cannot be expressed in a deterministic approach, due to uncertainty in the variables or the system structure itself, probabilistic reasoning approaches come into play. Due to their statistical nature, they fundamentally differ from the deterministic approaches, introduced before. However, they are becoming increasingly popular for system modeling as well.

Many different probabilistic modeling approaches have been suggested in the literature like fuzzy logic, Kalman filter, Artificial Neural Network (ANN) and Hidden Markov Model (HMM), however BNs - as will be shown in the next sections - are the most appropriate ones to describe relative influences among real world facts, especially if they are afflicted with uncertainty.

In Artificial Intelligence (AI) BNs are very popular since they are representing a generalization of HMM, which are only valid for discrete state processes. [174]. The dynamic variant of BN (DBN) have successfully been used for temporal reasoning [161] in various domains e.g. for speech recognition [247]. Their impact in medical domain and for physiological simulations will be discussed in chapter 3 - "State of the art".

In the following subsections, however, there will be a very short and summarized introduction

into the basic terminology and methods of probabilistic or Bayesian networks.

2.4.1 Definition

A BN is first and foremost a representation for probabilistic causality. Causal relations are given by a combination of a qualitative and a quantitative part [118].

The **qualitative part** is given by a graphical model represented specified by a Directed Acyclic Graph (DAG)[4] with nodes and edges. The nodes represent variables while the edges formulate the causal influence between the variables of the network (graphical model).

The **quantitative part** is describing the strength of a causal influence or relation between two variables. This strength is represented by conditional probabilities, which will be explained soon. Due to the stochastic nature of causal influences, the network is called Bayesian network. Other notations, as given in literature, e.g. probabilistic network, belief network, causal network are also used in different domains, but describe the same subject [118][128].

In probabilistic terminology, one is always considering random events and/or random variables. Unfortunately, the syntax is often mismatched and we need to clarify what is meant by our definition.

In formula 2.33 the Bayes' Theorem [5], which is the foundation of the BN theory, is shown. a and b are random events, and $P(a)$ and $P(b)$ are corresponding probabilities, while $P(a|b)$ describes a conditional probability.

$$P(a|b) = \frac{P(b|a) \cdot P(a)}{P(b)} \tag{2.33}$$

The key to connect events to random variables is given by introducing a finite number of ***mutually exclusive states***. Assuming A is a random variable with n states $a_1, ..., a_n$, then $P(A)$ is not a probability anymore but a probability distribution over all n states of random variable A. So $P(A = a_i) = P(a_i)$ is describing an event again, which represents the variable A to be a state a_i.

Although, in this case implicitly discrete and finite number of cases have been assumed, the equations are also valid for the infinite and continuous case. Thus in literature one distinguishes between ***discrete or continuous*** BNs.

[4] A graph that has no cycles.
[5] According to Thomas Bayes (1702-1761)

For the continuous case the Conditional Probability Distribution (CPD) is represented by a Probability Density Function (PDF), usually given by mixture of truncated exponential functions or the Gaussian distribution [43]. Of course, in the discrete case, the CPD is also a representation of a discrete sampling form a corresponding PDF, which does not need to be restricted to a Gaussian function. This is one reason why for BN modeling, one is normally using discrete nodes. Another reason is that the states of the nodes are usually representations for real world events. Therefore, it is common to use discrete states to represent different mutually exclusive events, providing a better mapping of real world phenomena.

For special continuous cases, e.g. Gaussian distribution, it is possible to mix static and continuous BN models to have a better modeling of the real world. The impact is obvious, if there are physical parameters, which are disturbed by Gaussian noise but have causal influence on other static nodes. In this case, one is speaking of a ***hybrid BN*** . Of course, one is always interested in reasoning from a BN and therefore the process of inference - which will discussed soon - is very important, based on how probabilistic influences are propagating through the nodes of a probabilistic network. For hybrid BN - due to the mixture of different models - one has to consider additional algorithmic constraints for inference calculation, e.g. discrete variables (nodes) cannot have continuous parents [138].

The most important rule for BN analysis is the *chain rule*, which can be motivated by the successive application of the fundamental rule:

$$\begin{aligned} P(X_1,...,X_N) &= P(X_1,...,X_{N-1}) \cdot P(X_N|P(X_1,...,X_{N-1})) \\ &= P(X_1,...,X_{N-2}) \cdot P(X_{N-1}|P(X_1,...,X_{N-2})) \cdot P(X_N|P(X_1,...,X_{N-1})) \\ &= P(X_1) \cdot P(X_2|X_1) \cdot ... \cdot P(X_N|X_1,...,X_{N-1}) \\ &= P(X_1) \cdot \prod_{i=2}^{N} P(X_i|X_1,...,X_{i-1}) \\ &= \prod_{i=1}^{N} P(X_i|X_1,...,X_{i-1}) \end{aligned} \tag{2.34}$$

Assuming that X_i $(i = 1..N)$ are representing N different random variables, and $\pi(X_i)$ are the parents of the variable X_i, which are given by the structure of a DAG, the CPD for X_i given its parents is given by $P(X_i|\pi(X_i))$. Then, according to the chain rule for BN, the joint probability distribution $P(X_1,...,X_N)$ can be reduced to the formula 2.35, which is the basic for Bayesian analysis.

$$P(X_1, ..., X_N) = \prod_{i=1}^{N} P(X_i|\pi(X_i)) \tag{2.35}$$

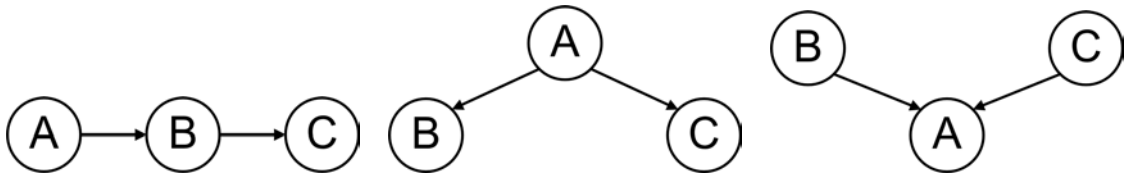
As one can see, the main part of the Bayesian formula is given by the CPD for variables X_i and their parents $\pi(X_i)$. This distribution is also known as a Local Probability Distribution (LPD). The main idea of the chain rule is to break up the joint probability into LPD, which is also known as *factorization*.

2.4.2 Structural And Conditional Independency

When talking about causal relations in a BN, we mean family relations (parent - child) given by the structure of the DAG. So if there is a link from A to B, we say that B is a child of A and A is a parent of B, especially $A \in \pi(B)$.

As shown in equation 2.35, the joint probability calculus is already reduced to the CPD given by the parents of one node or variable. The given factorization can additionally be simplified if there is a conditional independency, called **d-separation**.

D-separation can occur in three different structural cases, as shown in figures 2.10(a), 2.10(b) and 2.10(c).



(a) Serial connection (b) Diverging connection (c) Converging connection

Figure 2.10: Different connection types, where d-separation can occur: serial ((a)), diverging ((b)), converging ((c))

To describe the d-separation exemplary, we consider the serial connection 2.10(a). In this case the simplification is given by $P(C|A, B) = P(C|B)$, which means that C and A are *conditionally independent (d-seperated)* if B is given.

If we consider a cycle in the graphical model, which is depicted in figure 2.11, according to 2.35 the joint probability given by the factorization cannot be determined, because then one node could become his own parent. Therefore, cycles are forbidden for the graphical structure.

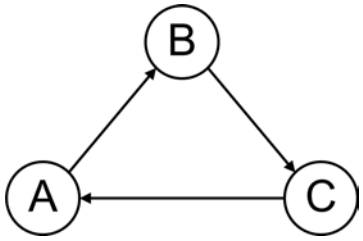


Figure 2.11: Cyclic graph. Cycles are forbidden for BNs, because the fundamental rule will lead into an infinite factorization process.

2.4.3 Conditional Probability Distributions

However, the qualitative part is given by the structure of DAG and has major impact on simplifications due to d-separation rules during the factorization process, the quantitative part is very important to describe the impact of relations in the network.

One can distinguish between continuous and discrete BNs, which means that the stochastic events, which are represented by the nodes of the network, can be described in continuous or discretized manner. This characteristic determines the description and form of the induced CPD functions.

In the continuous case, the CPD function is usually determined by a distribution function, which can be expressed by sum of exponential functions. Normally, for simplification reasons, Gaussian are used in this case. By discretization of the distribution function according to sample points, which represents discrete stochastic events, one can transfer continuous descriptions to discrete one. This process is called discretization, which is used in case of hybrid BN (static and continuous nodes together in one network). Of course, discretization can also be applied on learning or sampling data, which could exist in continuous form.

For the discrete case, the information about the CPD - which is discrete now - is therefore held in a Conditional Probability Table (CPT). On the one hand, one is not limited to distribution functions, which should be e.g. Gaussian and thus is more flexible. On the other hand, more information need to be kept in memory (in form of look-up tables) according to the **size** of the network, which leads to a *space complexity* problem for BNs. The size of a CPT for the node X_i is given by:

$$size(CPT_i) = r_i \cdot \prod_{j=1..N} r(\pi_j(X_i) \leq r_i \cdot r(\pi)^N \tag{2.36}$$

while r_i is the number of states for node X_i and $r(\pi_j(X_i))$ is the number of states for the j-th parent (Assuming N parents) of X_i. In the worst case, the size is exponential in number of parents. Therefore, it is very important to keep the number of parents as small as possible. A bad example is the Quick Medical Reference (QMR) [141] method (depicted in figure 2.12), which is used as clinical Decision Support System (DSS) to evaluate medical databases by connecting all diseases to all symptoms.

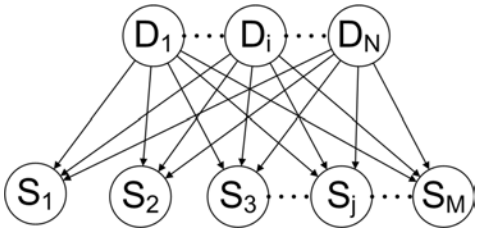


Figure 2.12: QMR network, an example for bad network structure. The size of the CPTs will rice exponentially in number of parents for this type of network structure. In praxis, the number of diseases $N \approx 600$ and the number of symptoms $M \approx 4000$.

Although, this network is not optimal considering the space complexity and could be optimized easily by reducing unnecessary arcs, e.g. for those connections, where no symptom is affected by a corresponding disease, nevertheless it is a good example for BNs used for practical problems. The direction of the arcs is certainly not arbitrary and always shows the way of causality, e.g. here a disease leads to several symptoms. This law is important for the information propagation through the network (see d-separation), even if the later usage of the network is inverse to the direction of arcs (in this example: if one knows the symptoms and wants to find out (reason) the disease with the highest probability to cause this symptoms), which is typically for ***diagnostic reasoning***.

2.4.4 Evidence and Reasoning

Observations are the key for usage of BNs by a process called ***evidence*** [118][128]. This means that due to observations of probabilistic events or variables, their state is known or certain. Corresponding nodes are called evidence nodes and their impact to other not observed nodes can be calculated, which is the process of reasoning.

Assuming that causality is defined by the relation between cause and effect in BN, three possibilities can be distinguished for probabilistic reasoning:

- Causal reasoning: The cause is known and one is interested in the effects. In the example from figure 2.12, the disease would be known and the network is supposed to provide

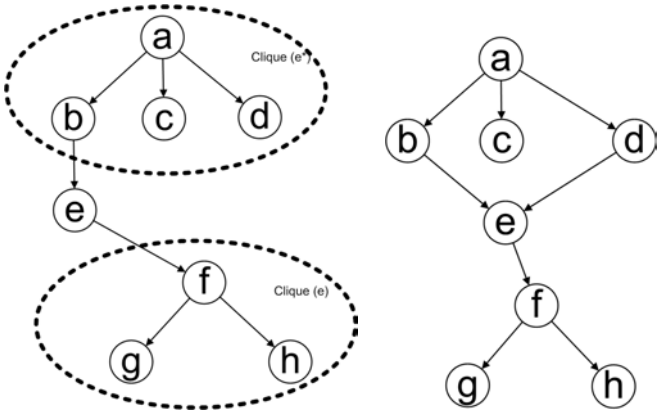
information about the symptoms (effects of the disease).

- Diagnostic reasoning: The effects are known and one is interested in the causes. In the mentioned example, this means, that the symptoms have been observed and the network is supposed to provide information about the diseases, which cause the symptoms.

- Combination of the other two. If one assumes more complex networks with more than two layers, it is possible that partially effects are observable as well as causal nodes, such that the knowledge can be combined.

Whenever an observation for a node is possible, evidence will be set to this node, which means, that the probability for the observed event is set to one. Then this information will affect other nodes in the network according to d-separation and marginalization, the process of ***inference*** can start and belief propagates through the network structure.

2.4.5 Inference



(a) Single connected BN (b) Multiple connected BN

Figure 2.13: Single connected ((b)) vs. multiple connected ((b)) BN: On the left side, the idea of cliques is depicted as well, considering the node e, the belief update is dependent on $Clique(e*)$ and $Clique(e)$. On the right side, the belief of node e is dependent on two different paths, due to the converging structure.

The process of inference describes the computational process to calculate conditional probabilities of nodes in the probabilistic network under the condition that evidence for other nodes has been set. There are various different methods existing here, distinguishing between ***exact*** and ***approximate*** inference techniques.

In exact inference, one is considering two subsets: evidence nodes, also called visible nodes

$Y_N = Y_1, ..., Y_N$ and not observed / hidden nodes $X_M = X_1, ..., X_M$, while $Z_L = X_M \bigcup Y_N$ is the set of all network nodes. Usually, one is not interested to know the effect of inference for all X_M, therefore lets assume $U_K \subseteq X_M$ is a subset and we are interested in $P(U_K|Y_N)$, which is given by marginalization:

$$P(U_K|Y_N) = \frac{P(U_K, Y_N)}{P(Y_N)} = \frac{P(U_K, Y_N)}{\sum_{U_K} P(U_K|Y_N)} \tag{2.37}$$

This method of exact inference is in general NP-hard [244][48], however for special topologies of BN structure (e.g. trees, single-connected nodes as depicted in figure 2.13(a)) more efficient algorithms can be found [161]. Additionally, especially for big networks, approximate algorithms (e.g. Monte-Carlo methods as Gibbs-Sampling) with lower complexity are very promising [43].

In general, for single connected networks (figure 2.13(a)), it is also possible to separate more independent parts of the network, so called cliques. By this the belief propagation can be encapsulated and many optimization techniques therefore are build on methods, that change the network structure to a single connected network or a junction tree [118] [244], based on methods from graph theory [50].

2.5 Dynamic Bayesian Networks

DBN are generally used to model temporal (dynamic) stochastic processes [71]. Dynamic does not mean that structure of the network changes over time. For simplification reasons, only discrete temporal processes are considered furthermore, although there are general approaches to model continuous time DBN - also known as continuous time Bayesian network (CTBN) [171]. However, discrete DBN have proven well applicability for modeling in many different areas of science.

A DBN is an extension of the BN formalism and the difference relies on *dynamic temporal changes* of stochastic events. By this definition, each variable X_i of the BN (see definition on page 42) for $i = 1..N$ independent variables is considered at different *time slices* $t = 1..T$ and is thus expressed by the notation X_i^t.

2.5.1 Definition

A DBN is defined as a pair (G, P), where G is a directed acyclic graph, which nodes correspond to a set of random variables X_i of a stochastic time dependent process $X = X_t^i : t = 1..T, i = 1..N$, where T is the total number of discrete time slices and N is the number of discrete random variables. P is then similar to equation 2.35 defined as Joint Probability Distribution (JPD) of variables of the random process X as shown in equation 2.38 [166].

$$P(X_1^t, ..., X_N^t)_{t=1..T} = \prod_{t=1}^{T} \prod_{i=1}^{N} P(X_i^t | \pi(X_i^t)) \tag{2.38}$$

$\pi(X_i^t)$ describe the parents from variable/node i at time slice t. One has to consider that parent nodes $pi(X_i^t)$ can exist in time slices different than t and usually are restricted to $t_\pi \leq t$, because usually no influence from future processes to past ones are assumed.

$P(X_i^t | \pi(X_i^t))$ is the conditional probability of variable X_i^t given the probability of its parents $\pi(X_i^t)$, which are specified by graph G, describing how far discrete variables are conditional dependent to other ones. The strength of the dependency, however, is given by the CPD, which in the discrete case is managed by finite CPT.

Also for DBN, the combination of both quantitative part (DAG) and qualitative part (CPT) is describing a probabilistic causal relation between nodes. The relation is usually the form, that the state of an effect node X_i^t is related to the state of cause node(s) $\pi(X_i^t)$ and vise versa, while the direction of the arcs in the DAG G is typically defined unidirectional from $\pi(X_i^t)$ to X_i^t (no cycles are allowed).

2.5.2 Unrolling

For each $t = 1..T$, one can consider a **time slice** of the stoachstic random process X, which is defined by a set of variables $X_t = (X_1^t, ..., X_N^t)$. Each time slice is defined by a DAG, while there are additional **dynamic arcs**, which define dependencies between different time slices. This is demonstrated in figure 2.14, while arcs of the DAG within each set (time slice) are represented by solid lines, and arcs describing the dynamic (temporal) dependencies are drawn with dotted lines.

The representation in figure 2.14 is known as the **unrolled** DBN view. As one can see, the DBN - in unrolled view - is nothing else than a complex BN with $T \cdot N$ nodes instead

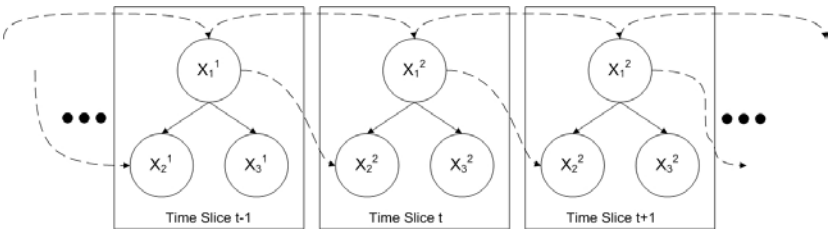


Figure 2.14: DBN with representation by time slices, dotted lines describe dynamic temporal dependencies, while solid lines are valid within each time-slice

of only N. Usually in praxis mixtures of BN and DBN are used, which means that some nodes are independent to the observation or inference at different time slices. Such nodes are non-temporal BN nodes and others are temporal DBN nodes. Due to simplification and a more intuitive representation, a terminology introduced by Decision Systems Laboratory (DSL) [66] is used furthermore, which is depicted in figure 2.15, representing a mixture of BN and DBN nodes. Here, two different views are visualized, on the left hand a compact representation and on the right hand an unrolled view of the BN/DBN mixture. In the compact (rolled) graphical representation of the mixture, the dotted lines describe dynamic (temporal) dependencies, while the number describes the depth of the relation. The solid lines represent static dependencies. Additionally one can see the double framed node Y_1, which represents a constant node, showing the independcy from temporal observations. Each DBN/BN mixture can be unrolled to a BN graphical representation. The analogue unrolled BN is depicted on the right hand of figure 2.15 for the case of $T = 2$ total time slices. In general, due to the transformation from DBN to BN, all known algorithms known from the field of BN for learning, belief propagation and inference can also be used for BN. One has to consider that the size and number of the CPTs is additionally increased by the number of time slices T.

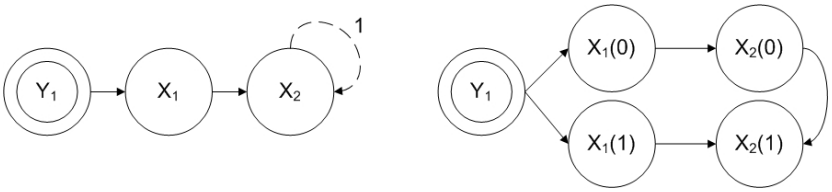


Figure 2.15: BN DBN mixture, rolled (left) and unrolled network (right): The representations on both hand sides are equivalent. Usually, the compact representation is used for the design phase and the unrolling is important for algorithmic processing. The dotted line on the left describes a temporal arc and the number is representing the temporal depth. The unrolled equivalent is a solid arc between nodes $X_2(0)$ and $X_2(1)$. Y_1 is a static node, independent from temporal observations.

3 State Of The Art

In the following chapter the *actual* state of the art (January 2010) will be considered. In each section the relevance, analogy or differences to the actual work will be discussed.

Initially, there will be an overview of ***simulation devices***, considering the broad range starting from computer-only (screen-based or virtual) over specialized (part-scale) to full-scale ***simulators*** [49]. This is a very important section, because this work was highly motivated to find applicative usage in real systems. The state of the art shows, especially, the recent developments as well as problems and weaknesses of current medical simulation systems. The focus thereby will be on the role of physiological simulations and the models and methods applied in educational simulators.

In the subsequent section, there will be a review of ***physiological simulation methods and models***, typically with regard to circulatory, respiratory, and electro-physiological systems, which play an important role for medical simulation and teaching systems.

Due to the importance of ***pharmacological simulations***, also for educational devices, they will be considered in a separated section, while the priority will be on Physiologically Based Pharmaco-Kinetics (PBPK) and Physiologically Based Pharmaco-Dynamics (PBPD) modeling approaches.

Various methods, systems and tools already exist to provide a platform for physiological modeling and simulation. Therefore in one section different ***simulation systems and description languages*** will be compared and will be pointed out, why an independent tool was developed during this work.

Then, static and dynamic ***Bayesian networks*** from the domain of stochastic methods for modeling will be introduced and discussed in detail, including applications, algorithmic methods for learning and inference and practical tools and problems. The various aspects will also be highlighted in comparison to medical applications and physiological systems.

And finally different ***simulation system architectures*** will be compared regarding their

usability for set up simulation systems in physiology. A concluding ***Review*** will summarize the important aspects and highlight the role of this thesis.

3.1 Simulation Devices And Full Scale Simulators

Most medical simulators are originally from USA, but also in Europe simulation systems - especially those for anesthesia interventions - are increasingly established in medical education [86][49]. The main idea is to facilitate medical skills by providing medical scenarios as realistic as possible by highly integrated artificial patients [177].

Whether a simulator is designed to fulfill only one task (task-specific simulators, part-scale simulators or task trainers) or is capable to educate in different directions by covering multiple tasks, all simulators focus on to be as close as possible to reality.

3.1.1 Screen Based Simulators

Screen-based (computer only) simulators have on the one hand the bottleneck to not cover all sense perceptions, especially haptics. Therefore, they usually have limited realism, but are nevertheless very important for medical education, since they are specialized on more detailed and complex computations in comparison to full-scale systems [86]. Additionally, they are much easier to use, due to their mobility [134] or the implementation with web-based front-ends, e.g. the Virtual Anesthesia Machine [229] or the Web-Human - a website physiology teaching simulation [239].

Thus one can distinguish between the impression of realism which is given externally by expensive and complex hardware (emulators, see also chapter 6) or the exactness of the simulation, which is more related to the complexity of internal used models. This two aspects are combined within this thesis.

Many simulation software systems - most of them have their origin in anesthesia education - are known as pioneers of the medical simulators. A good example is GasMan [78], a Graphical User Interface (GUI) based simulation software for anesthesia training, which was found 1983 and is permanently further developed up to now providing a very effective tool for goal-based learning of special skills [181]. Also well-known is Body Simulation [26], which includes respiratory and circulatory simulations, and although found 1973 by Smith et al., it is updated permanently and very helpful in understanding of physiological or pharmacological effects in

anesthesia [211].

However, many screen-based simulation tools simply vanished after their development. E.g. VentSim [199], a simulator for cardiopulmonary physiology or HUMAN [45], one of the first physiology software simulators (1983) written in Fortran, is as unknown as - the for those times popular - operating system CP-M80, which was used.

This is usually because most systems are very restrictive, too specific and limited and do not provide flexibility to change, upgrade or extend models, that is a new feature of systems, based on simulation description languages as e.g. given by the active European Physiome Project [69][182][108], or other efforts to establish a general modeling language format, which could then be independent of systems and environments [103].

Nevertheless, all the pioneers - whether they still are used or not - and the new simulation tools [15], have shown their effectivity in improvement of medical education [134] and medical skills [46][122].

Due to the accuracy of screen based simulation systems considering **complex physiological interactions** [86] e.g. PBPD or PBPK, recently many other part-scale or full-scale simulator systems are trying to combine both systems together [205]. By the integration of complex physiological simulation to the a task-trainer, the trainee could also control his or her knowledge by comparison with the physiological simulation. Additional debriefing, which is possible by quantitative simulations, improves the mannequin based teaching[205].

One of the famous and recent screen based simulators is the Maryland Virtual Patient (MVP) [117][156], which is especially focusing on role-based diagnosis and treatment of virtual patients, enabled by the help of a virtual tutor [170]. The key components of this system are on the one hand the knowledge-based representation of health and disease and on the other hand the cognitive and psychological impact to health states, which is yet unique among all medical simulators (see chapter 7 for more information). The simulation is essentially using Natural Language Processing (NLP) with expertise of Nierenberg et al. [206] in form of ontological recorded complex chains [155].

3.1.2 Fully VR Simulators

A 3D environment to deal with medical situations by using avatars is **Sim-Patient** [196], widely used for military medic training. Such systems integrate the profound knowledge from game technology into medical education and training simulators.

However, clinical procedures are increasingly simulated in virtual reality, because the mechanical simulations are too expensive and by this, show another trend for future simulator developments. A good example is given by the virtual reality simulator for hip surgery [25], developed in 2008.

University of Auckland - Virtual Medical Center - offered on second life, is an attempt to simulate a full hospital environment within a VR framework (see fig. 3.1). Other environments known as *Second Health* [98] (see figure 3.2) are becoming increasingly popular, due to the well 3D fundament of Second Life and the easy web-based access for all attendees. In all this environments, a virtual patient can be treated by the avatar physician for all phases of medical treatment, e.g. anamnesis, examination, diagnosis and therapy.

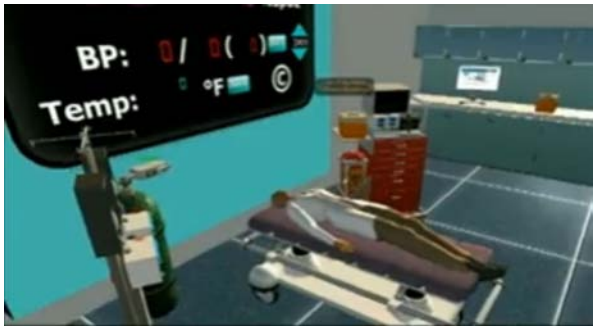


Figure 3.1: University of Auckland - Virtual Medical Center: Medical equipment can be attached to the patient to help the diagnosis of diseases, e.g. by monitoring of blood pressure and temperature.

Up to now, these simulators focus on anatomy, 3D visualization and graphics effects. If simple physiological processes are integrated (as shown in Fig. 3.1), they are based on very simple script formalisms and do not mirror complex integrative physiology yet.

Although, an increasing number of simulations contain 3D effects and VR methods, fullsscale and task-specific simulators with mechanical emulation - which could also integrate VR methods (resulting in augmented reality) - are still not only popular but due to their realism, they also achieve much better training values. Therefore, the increased trend toward fully virtual medical environments and simulations has to be questioned [62]. However, all famous mannequin simulator companies have virtual presences in Second Health, providing basic virtual teaching knowledge already.

Figure 3.2: Second Health - Simulated Hospital: A patient can be treated virtually by an avatar physician during all phases of medical treatments, including anamnesis, examination, diagnosis and therapy. Additional diagnostic aid is given as well by virtual medical equipment, e.g. ECG device or medical imaging.

3.1.3 Full-Scale Simulators

The first medical full-scale simulator - called "Sim-One" was invented in the late 1960s by the group of Prof. Stephen Abrahamson [59][163]. This historical simulator (illustrated in figure 3.3 on page 54) was ahead of one's time and was including many different simulation components, which are typical of modern full-scale simulators. For instance, it could simulate the influence of four different medicaments into the physiological feedback including side-effects, heartbeat and pulse, as well as lifelike skin and teeth.

Unfortunately, the control unit was as big as a garage. Since that time, the different mechanical and electrical components of the simulators have become increasingly smaller. Although the medical community's initial response to Sim-One was skeptical, nowadays, medical simulators are used in wide range for teaching medical skills, and hospitals around the world are using computerized human patient simulators to train medical personnel.

The key idea for designing a full-scale simulator is to provide a very **realistic** simulation. Research has always proven, that the learning effect in realistic environments is significantly higher [49][62]. Due to their realism, full-scale simulators do not only provide a multi task-specific education, they are also important for risk management, e.g. in anesthesiology [143].

Currently, full-scale human patient simulators are commercially available in several versions from (among others) two main manufacturers (METI [159] and Laerdal [133]).

The Human Patient Simulator (HPS) from METI [159] is one of the most popular physiological patient simulator. It has been used for military as well as for civil educational programs and integrates many different physiological components. The applications range from simple task-

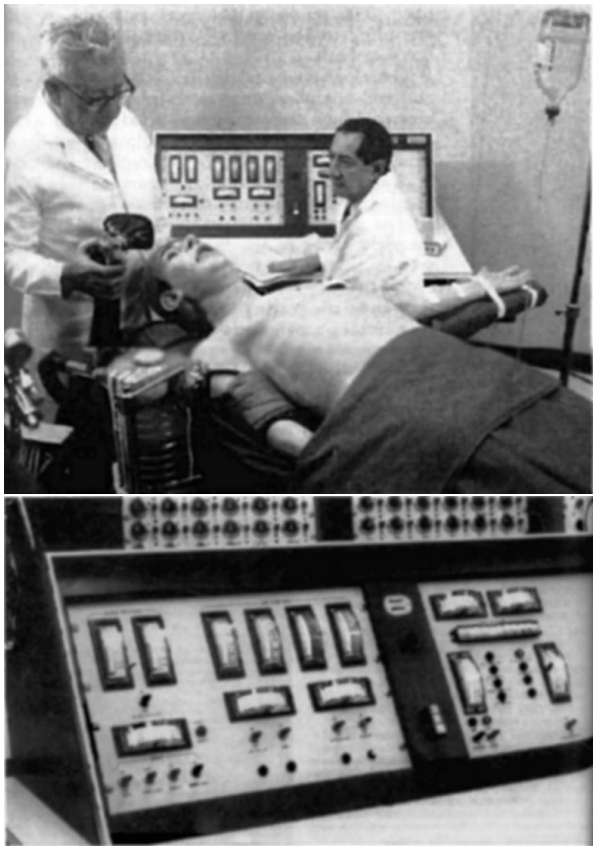


Figure 3.3: Sim-One: The first full-scale simulator was using a control unit as big as a garage.

specific simulations (e.g. ECG learning) to complex surgical ones.

HPS is especially present in medical simulation centers and provides specialized educational programs. A good example is given by the "Heidelberg Anaesthesie- und Notfall-Simulator" (HANS) [100], which includes circulatory, respiratory, metabolic, neurological simulations and emulations (e.g. pupil reaction) relevant for anesthesiology in a very realistic way.

However, this simulator is so complex to use and control, that always specialized administrators are necessary to perform a simulation run. It is not an "off the shelf" teaching device [194].

Considering the full-scale simulator SimMan (the advanced simulator from Laerdal [133]), depicted in figure 3.5, one can see the tendency for integration of physiological simulations into full-scale simulators as well. SimMan has separated the mechanical and hardware parts of the manikin from the software part of the system.

Here, time varying physiological parameters (12-lead ECG and trends) are only simulated screen-based on an artificial patient monitor. By this method, feedback and debriefing is

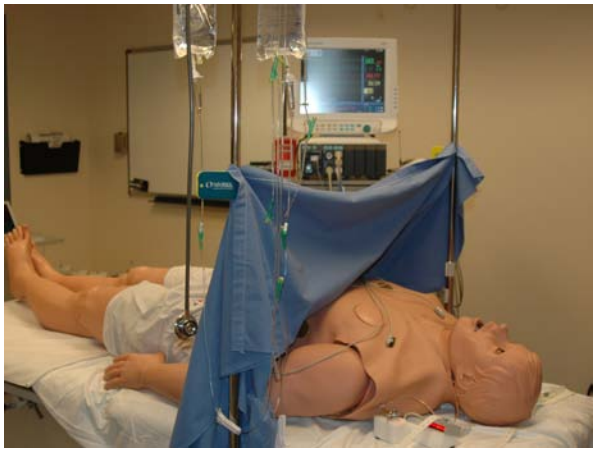


Figure 3.4: HPS - Human patient simulator system from METI

Figure 3.5: SimMan: full-scale simulator with monitors for vital parameters and physiological monitoring. The manikin (hardware) is separated from virtual signal simulation (software).

easily integrated into the software system. Although the airway management of the SimMan is emulated in hardware, other important physiological parameters (e.g. ECG or blood pressure) are not emulated and thus could not be accessed by standard monitoring systems.

Considering the physiology, the Laerdal software is using ***"trends"*** for physiological parameters (e.g. HR, Temperature, respiratory rate, blood pressure and other). These trends are essentially described by static predefined curves, allowing an easy development of scenarios on the one hand (SimMan Trendeditor). On the other hand, trends possibly have nothing to do with a real physiology. A drug reaction - described by a trend - is not a physiologically based reaction, because it is static and independent from patients current state.

In summary, the **models behind** the simulation show many limitations. Laerdal's SimMan does not have physiological or pharmacological modeling, but uses programmable scenarios and the overlaying of physiological trends (eg. changes to heart rate, blood pressure or respiration).

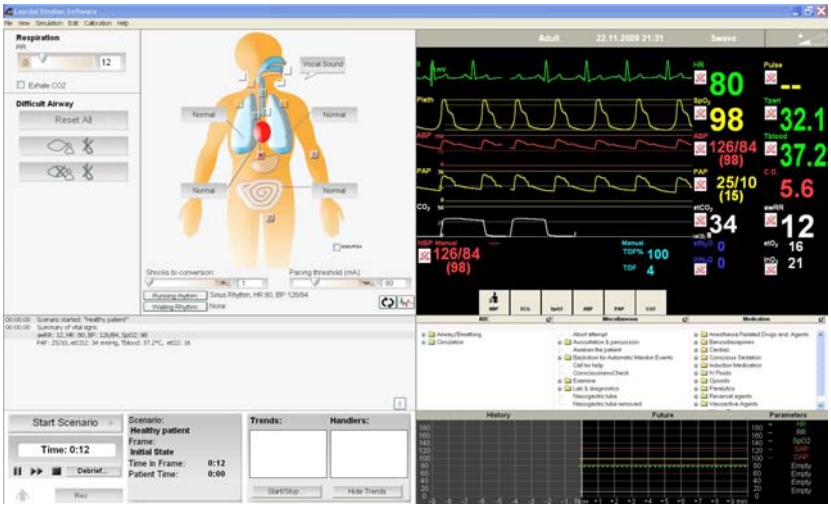


Figure 3.6: SimMan software: Physiological trends are used to simulate medical scenarios. The resulting simulations are therefore encapsulated from real physiological models and are based on static data.

This is again difficult and inconvenient, because scenarios and trends have to be written and validated by specialized personal and methods.

METT's HPS does provide pharmacological and physiological models indeed, but due to professional experience, these models are not predictable or consistent with published evidence or clinical experience[194]. They require administrators and instructors, who are familiar and confident in the use of the METI HPS control, to edit or design the models in an adequate way.

In terms of teaching clinical pharmacology, the simulators do not use complex and accurate models. The learning value for student trainees lies in the correct choice of drugs and not the exact value (in milligram). The learning effect from the patient assessment as well as the simulation scenario and the link between cognitive and behavioral training is crucial [194].

Part-scale or screen-based simulators can be applied very effectively for typical basic teaching skills and for goal-based learning, such as respiratory physiology and cardiovascular haemodynamics [84]. But if one is interested in advanced clinical skills, e.g. management of difficult airways, tension pneumothorax, pulmonary embolism, as well as shock and emergency management, e.g. for Anesthetic Crisis Resource Management (ACRM), only full-scale simulator systems provide the capable realism and allow the necessary interactions (e.g. team management, communication, planning), which is impossible to realize in fully virtual environments yet [76].

3.1.4 Special Purpose Simulators

Although special purpose (goal-based) simulators normally focus on very specific areas of medical education (e.g. for surgery [95], catheter interventions [158][37] or rescue management [133]), there is a general request to have an integration of such systems into full-scale systems. Additionally, the physiology and pharmacology are becoming increasingly important in such systems as one can see for SimMan and HPS.

The endovascular interventional simulator VIST-C [158] is e.g. integrating 12-Channel ECG and catheter based blood pressure monitoring by a screen-based simulation. This simulation of vital functions is virtually and can be compared to SimMan (Leardal [133]).

Another European part-scale simulator is the Cathi system [37][190]. The motto of Cathi is to provide a simulated patient, which should feel like as real as possible. Therefore, in the Cathi system (see fig. 3.7) the intervention is done by using original instruments (e.g. catheter, balloon). Cathi provide force-feedback (haptic sense) and support a working environment similar to the catheter lab itself.

Although - also here - simple vital parameters as ECG are provided, or simple pharmacological mechanisms are existing for different drugs, a substantial improvement can be achieved by integration of a physiological modeling system and emulation of vital signals, which are aimed for future, supported by this work.

Also for the class of special purpose simulators the fundamental strategy question is, whether to focus on script-based or model-driven simulations [232]. The trend, however, is going toward ***model-driven*** (in contrary to script- or trend-based) simulations, because these are based on real physical phenomenon and thus lead to more realistic results. Simulator systems like Cathi, thus investigate more into physical based models and therefore their users and customers usually say - surprised - "It is so real"[1]

3.2 Physiological Models

As pointed out in the introduction, the terminology ***physiological modeling and simulation*** is overlapping many different areas of science with different focal levels, e.g. macroscopic (tissue, organs), microscopic (cellular) or even nanoscopic (metabolic activities, molecular) levels. [57] [108].

[1] From the author's experience in the EuroPCR congress 2007 in Barcelona.

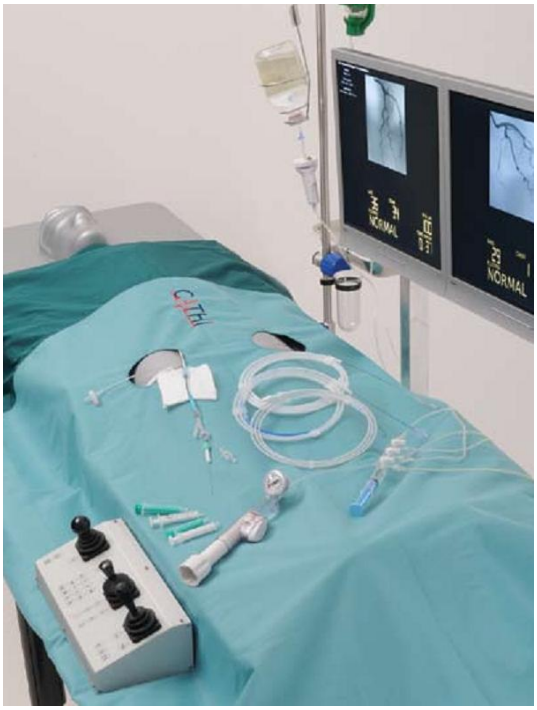


Figure 3.7: Cathi simulator, applied here for simulated interventional cardiology. Integration of a physiological modeling system, including cardiovascular system, drug interactions, vital signal emulation, is a substantial improvement for future task-specific simulator systems.

Nevertheless, it is the overall goal to - one day - unite all this different views and levels in a global description (European ICT health care initiative, virtual human patient) [182][47][110]. At the moment, these levels and the corresponding models are mostly separated, especially because the ***complexity*** of the resulting bigger models is too high for current computational systems [57].

As have been shown before, different simulator systems claim to provide physiological simulations, however all of them differ from each other in the level of modeling complexity. The differences are given by using integrative systems instead of predefined trend lines, as well as the level of detail and number of patient vital parameters.

Therefore, in this section the focus will be on those models and simulations, which are related to the overall topic of this thesis and mainly consider the modeling of circulatory, respiratory and electro-physiological system. The emulation devices are targeting blood pressure (IBP), oxygen saturation (SaO2) and electrocardiography (ECG) and are based on corresponding physiological parameters. Pharmacological simulations which are also often related to physiological parameters will be considered in the next section separately.

3.2.1 Cardiovascular System

Especially for haemodynamics of the heart and circulatory systems many different models have been proposed in the literature, describing relations between pressure, volume and resistance [14][79]. A good introduction and overview is given by the book "Mathematical Physiology" [125]. The basic strategies are presented in chapter 2.

Early CVS models can be found by Guyton et al. 1963 [90] [89], 1975 by Downey and Kirk (about coronary blood flow) [64][213] and 1985 by Arts and Reneman (myocardial circulation) [213], providing basis techniques for understanding of physiological modeling by equivalent hydraulic or electric descriptions.

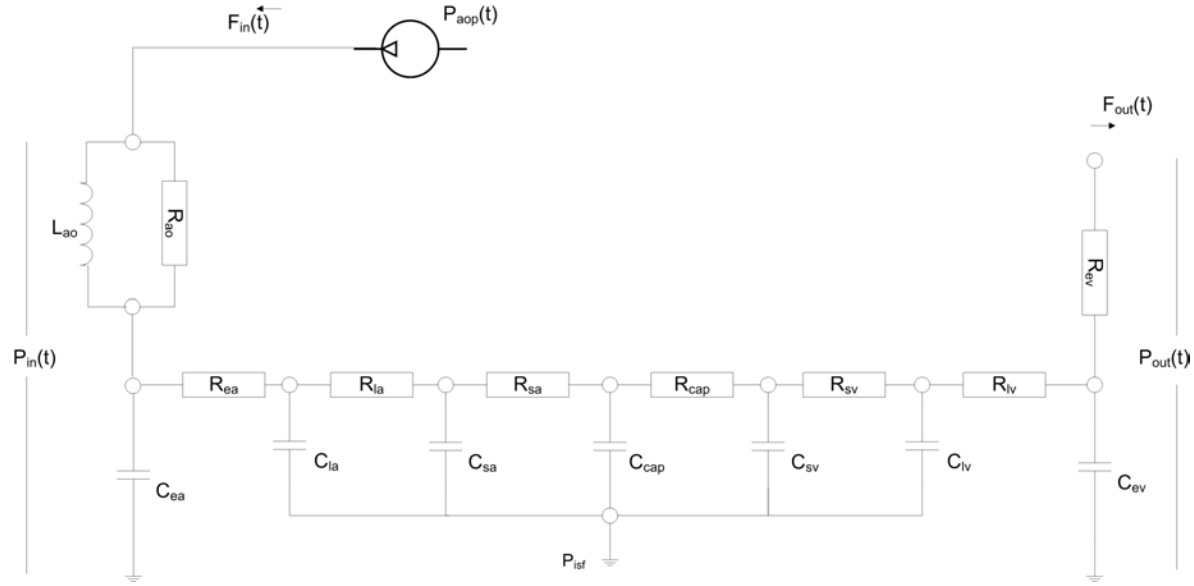


Figure 3.8: Coronary circulatory system based on Zinemanas et al. 1994 [246]. R are resistances, C compliance elements and L is an inductance. The input pressure $P_{in}(t)$ is represented by the generated pulsatile aortic pressure $P_{aop}(t)$. Following segments have been considered: aorta (ao), epicardia arteries(ea), large (la) and small arteries (sa) and capillaries(cap), small veins (sv), large veins (lv), epicardial veins (ev). P_{isf} is the pressure of the interstitial fluid. The output pressure $P_{out}(t)$ is equivalent to the pressure of the right atrium.

The models vary especially in the degrees of complexity (e.g. open and closed loop [42] [172]) or the focus to the diseases or dysfunctions [209], each model is interested in. Many models also only focus on parts of the cardiovascular system, e.g. the circulatory, pulmonary, systemic or the myocardial circulation.

While the first models were kept very simple and are based on Guyton's famous circulatory

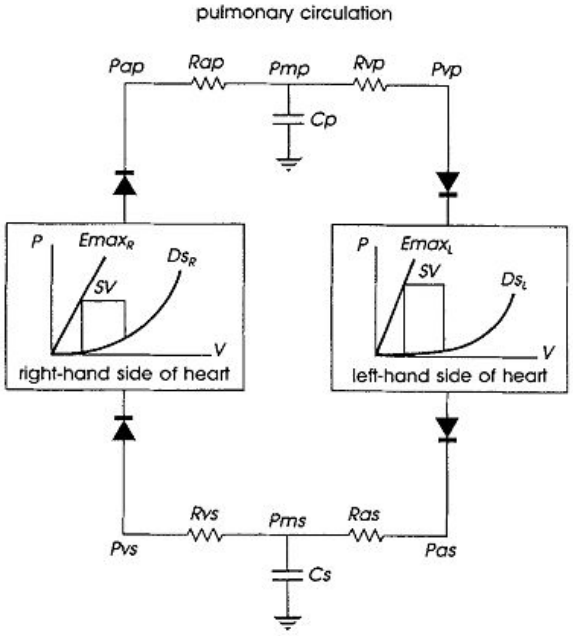


Figure 3.9: Simple closed loop CVS, used to model heart failures and anomalies. The dynamics of the heart chambers (essentially the ventricles mechanics) have been modeled using relations of the end-diastolic and end-systolic pressure volume relations and the heart stroke volume (SV) or the cardiac output (CO) . The top part of the figure is representing the pulmonary circulation, by the pulmonary arterial and venous resistances (R_{ap}, R_{vp}) and the overall pulmonary vascular capacitance C_p. The lower part is complementary representing the systemic circulation, by the systemic arterial and venous resistances (R_{as}, R_{vs}) and the overall systemic vascular capacitance C_s.

model [2] [90] [88], in the end of 1990s first attempts to extend existing models started. The reason for this lies in the impact of Guyton's model for the scientific community and the understanding of CVS physiology, which is mainly valid up to now and serves as basis for recent research as well [164] [91] [223].

The physiological modeling process - started after 1990s to incorporate different functional relations of the CVS into more complex models - is permanently further developed up to now. A good milestone was achieved 1994 by Zinemanas et al., introducing a mechanical model, incorporating functional relation of active myocardial mechanics and demonstrating interactions with a larger coronary circulatory system model [246] (figure 3.8).

[2] Arthur Guyton (1919-2003) is epically famous for his experiments studying the physiology of cardiac output and its relation to circulatory system. Guyton was the first, who proofed, that oxygen is regulating the CO and not the heart itself [94].

One of the first quantitative and mathematical formulated closed-loop models (depicted in fig. 3.9) for the CVS mechanics, was also given 1994 by Tsuruta et al., including very simple (two resistive and one compliance element) pulmonary and coronary circulation systems, which are connected by a simple representation of the heart chambers [226]. This model was originally used to quantitatively describe different heart failures and anomalies.

Autoregulation is another important aspect for CVS simulation, when closed loop systems are considered. Various approaches exist for this sub-domain too. The common method is to map the regulation process as near as possible to the real life process by mathematical functions. E.g. by using end-systolic and end-diastolic pressure volume relationships [226] or describing the interaction of carotid baro-receptors and the pulsation of heart within a mathematical baroregulation model [227].

Other approaches also consider information from the respiratory system into the autoregulaton process, e.g. by taking the partial pressure of carbon dioxide into account [222], which leads to a closed loop regulatory model for the cerebral blood flow.

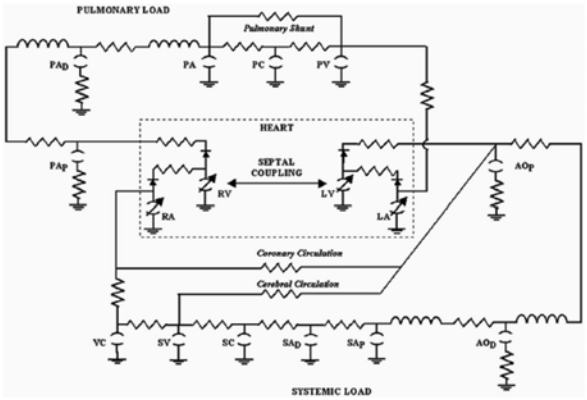


Figure 3.10: Closed loop model including a detailed pulmonary circulatory system. The model integrates new aspects like the modeling of shunts (e.g. pulmonary shunt at the top of the circuit), coronary circulation, cerebral circulation and septal coupling [172]. Following segments have been considered Aorta (AO), systemic and pulmonary arteries (SA, PA), systemic and pulmonary capillaries (SC, PC), systemic and pulmonary veins (SV, PV), vena cava (VC), right and left atrium and ventricle (RA, LA, RV, LV), while distal and proximal distinctions have been done as well (e.g. pulmonary arteries distal PAD).

In 2000, Olansen et al. extend the existing formalism and provide a detailed closed loop model. By this also a model for the pulmonary circulation was provided (see fig. 3.10). Together with the coronary circulation model [246] an integration into a overall systemic circulatory model was possible, which allows a more detailed view than the model given by Guyton [88], both sub-systems were merged by integration of heart into the model, describing

ventricular interaction [172] similar to the approach from Tsuruta et al. [226].

During the last years circulatory models have become more complex and detailed. In 2007 detailed haemodynamic simulations of the cerebral blood flow have been used to model cerebral vasospams [147][107], by applying the same principles as mentioned before, but changing the focus to the cerebral ciruclation. The same year a model was introduced by Smith et al. for the description of the regulatory effects of the Autonomic Nervous System (ANS) - including baroreceptors - [209] to the CVS, e.g. to Cardiac Output (CO). By this method one is able to model complex interactions e.g. during a septic shock. The strategy is depicted in figure 3.11.

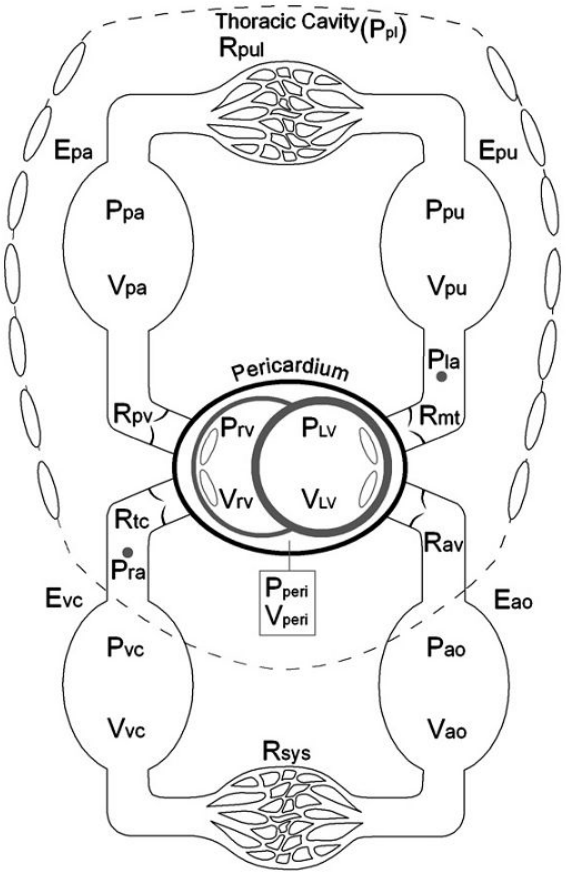


Figure 3.11: CVS with baroreceptors [209]. The minimal model from figure 3.12 is extended with regulatory processes.

Closed loop models, integrating regulation mechanisms of cardiovascular system, are becoming more and more interesting, due to the simulation advantages they offer. The goal of such models is to offer simulation and parameter estimation methods to help clinicians diagnose CVS dysfunctions [209]. Using the above mentioned hydromechanic simulations and combin-

ing the results with knowledge about mass transport in different organ compartments (PK), one is using the information to describe the changes of drug concentrations in human body in PBPK [191].

In summary the goal and effort in the previously mentioned literature is - and probably future work[3] will be - to use quantitative simulations from the cardiovascular system for reasoning [71]. Especially one is interested to explain diseases (dysfunctions) and thus to be capable to reason and detect shunts [172], heart failures [226], a septic shock [209] or spasms [147] by external measurements (real data) and comparison to simulation data. This objective has been followed in chapter 9.

By increasing the model complexity, it is more difficult to match the model parameters based on real data measurements, additionally problems with stability and convergence of the fitting algorithms have been reported [210]. Therefore a full ***minimal model*** for CVS was given by Smith et al. 2004 [210], which is clinical relevant, because it focuses on computational time efficiency and a minimalistic descriptions.

The model is depicted in figure 3.12 and is, as one can see, based on only six particular components. These components can be represented by compartments too, while additional mathematical regularization is used to control the values of the resulting system. The presented model is describing a closed loop CVS.

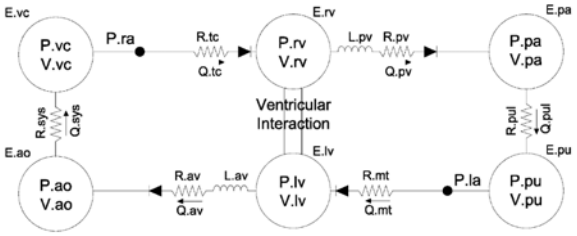


Figure 3.12: Minimal CVS presented by Smith et al. only applying six different components [210]

3.2.2 Respiratory System

In a respiratory system the idea is to model quantitatively oxygen (O_2) and carbon dioxide (CO_2) transport and exchange and consider effects to CVS. Normally (multi-) compartment models are applied for this purpose [33][125]. A good example is given by the distributed model

[3] Based on the developed work, the author has planned to use the proposed system for cerebral haemodynamic simulations in cooperation with the Mannheim Medical Center for better prediction of cerebral vasospasms [147] in near future.

from Carlson et al. for O_2 and CO_2 transport and exchange between a three compartment lung model and a pulmonary circulation model (blood tissue exchange) [36].

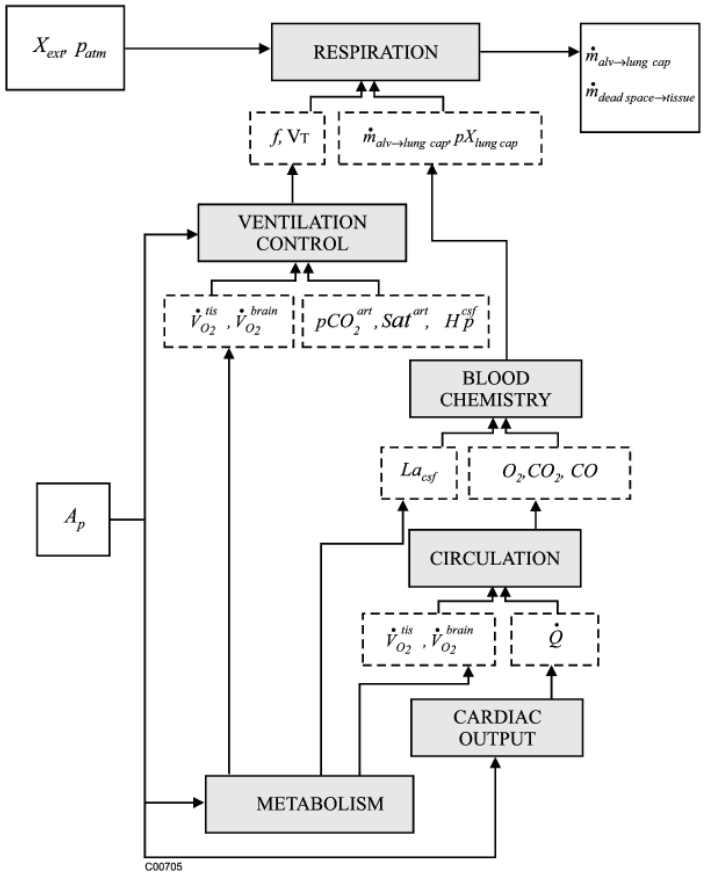


Figure 3.13: Ventilation response : A comprehensive mathematical model describing the relations between respiration, circulation, oxygen metabolism, and ventilatory control [216]. $\dot{Q}$ is describing the total blood flow. $\dot{V}_{O2}^{tis}$ and $\dot{V}_{O2}^{brain}$ are the tissue and brain oxygen consumption. $\dot{m}$ describes the mass transfer, A_p the physical activity level. The ventilation control component describes the influences to the breathing frequency f and tidal volume V_T.

One of the first respiratory simulations was performed on the Nottingham Physiology Simulator (NPS). Results could be used for predication of changes in mechanical ventilation [96].

An advanced model, including respiratory regulation, is presented 2005 by Stuhmiller et al. [216]. The model outline is is depicted in figure 3.13. As one can see, this model already includes complex regulatory mechanisms between ventilation and circulatory control.

However, for this thesis the mechanical ventilation is not important, because the simulator is not meant to support any airway management, there are various models existing for simulation of ventilation, e.g. [184], as well. The focus here is to model the pneuma-mechanic attributes,

similar to the approaches from haemodynamic modeling of the CVS, using known parameters like pressure, resistance, compliance and flow.

For this work the knowledge of the oxygen saturation in blood is important for the simulation and emulation of pulseoximetry [228] (Saturation of Oxygen (SaO2), which can be easily modeled by using compartment models [216] (See also chapter 2 - Basics).

3.2.3 Electrophysiology

Various methods are known to describe electro-physiological phenomena at different levels of detail, e.g. in cellular level [144] or in higher levels [200] given by the electrical field abstraction resulting in the ECG signal [125]. A good summary of the different methods was delivered 2008 by supervision of the author [195].

For medical education, however, modeling of ECG is important, because this is a standard in accessing patient vital parameter. One general way to model ECG is by assuming electrical sources and calculating the electrical propagation through tissue. The models differ in the way or number of this sources [200], e.g. point based, surface based (equivalent surface source model [233]), single and multiple dipoles.

In addition to the source, the conductivity needs to be described as well, which can be done in terms of surfaces, volumes or tensor fields [195]. One can use different techniques, e.g. finite element method, to solve the model equations, finding the surface charge density on the thorax.

This process is known as the ***forward problem*** of the electrocardiography, however the ***backward problem*** - finding (fitting) the parameters of the sources and conductivity models - is an inverse problem [207]. For the latter usually modern regularization techniques are applied to guarantee or accelerate the convergence of fitting algorithms [119].

One good and simple representation of electrical activity of heart is given by an electrical dipole model. [200]. Additionally the dynamic changes of the electrical activity is modeled by a system of linear differential equations [157]. This theory was adapted in the PhysioToolkit ECGSYN Software from PhysioNet research [183] to artificially describe and simulate dynamic ECG signals. The idea of the model is depicted in figure 3.14.

This model was extended during this thesis to a multi source representation by a ***multiple dipole model*** [5], which will be explained in more detail in section 6.4 - "ECG Emulation

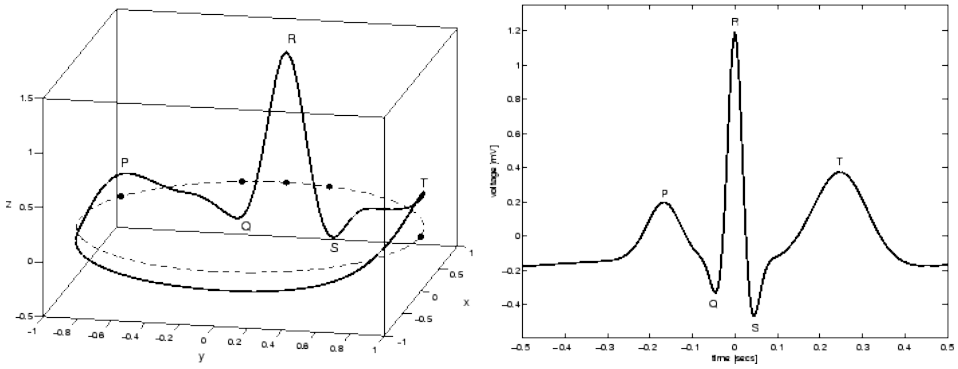


Figure 3.14: Dynamic ECG model : The ECG three-dimensional dipole (left) is described by a system of differential equations, containing the information about important markers e.g. the P or T wave and the QRS complex. The resulting ECG deviation signal (right) is calculated by the forward solution of the electrocardiography for the far-field and is essentially given by a dot product [157] [183]

Interface".

Other approaches - so called volume models - consider the propagation of the electrical activity induced by potential difference e.g. form the sinus node. For these methods a detailed 3d-model of the heart is necessary as well as the orientation of fibrils. [234]. Of course the latter model is more precise in modeling electrical heart activity, if the orientation and density of fibers are known. The latter are normally taken from histology data.

The actual efforts are concentrating on extraction of the knowledge about heart fibers, electrical activation centers (e.g. the sinus or AV node) from imaging modalities to forecast the electro physiology in case of manipulation (e.g. by a pacemaker) by functional imaging [19].

However, for the current work, the task to provide a quasi real ECG simulation and emulation system, is fully satisfied by providing dipole (source) based models, which can be solved forward and backward and thus can be learned from real patient data.

3.3 Physiological Based PK And PD Modeling

Another important field of physiological simulation, which is concerning reaction and diffusion of drugs, will be considered in the following. Various detailed models exists in this field, for example fine granulated models including metabolic reactions for molecular level, as well as simple but efficient macroscopic descriptions.

3.3. PHYSIOLOGICAL BASED PK AND PD MODELING

In the beginning of PK and PD modeling, usually only simple formulas (e.g. compartments) have been used to fit the observation of drug mass transport and reactions by population modeling or multivariate approximation. The reason for this is based on the big gap of information about the real complex physiological phenomena. During the time, however, the knowledge about drug metabolism and physiological behavior grows (although still many parameters are uncertain and non-observable) and thus the models could become better and more patient relevant.

The PK and PBPK modeling differ in the modeling strategy, the first is modeling the drug mass transport, without considering the real physiological information provided by the living individual, e.g. blood pressure and flow information. Such models are usually applying many simplifications and the model parameters are fitted to populations [125].

Population PK was the method used to correlate clinical data according to standard dosage regimens administered to patients, by considering variables as age, sex, weight and disease state [2]. By this, unfortunately, patient specific information could only be recognized in the variability (standard deviation) of the fitted population parameters (mean values) and got lost.

Both approaches can be tracked back to compartment modeling introduced by Torell [220] 1973, which was essentially based on the Rutherford's radioactive disintegration chain, described by compartment equations [193]. Since the general and basic developments and principles (presented in chapter 2 - "Basics") more than thousands of papers were published describing particular (PB)PK models [1].

Although compartment models provide a very flexible, well known and abstract representation for PK or PD modeling, they are also limited in certain ways . For instance they force the modeler to apply abstractions by introducing homogeneous compartments, where the real physiology is not homogeneous. Also the first order linear description can be a constraining assumption, because especially in case of instabilities of physiological systems, many higher order aspects or non-linearities are relevant [1].

By trying to consider exceedingly many physiological factors (e.g. blood flow, organ size, tissue partition coefficients), the models become more complex and the strategy turns toward patient individual PBPK modeling [68]. Clinical aid is supported for many pharmaceuticals by various software tools, e.g. the full body PKPD simulation software PK-Sim [81]. The main principle is to connect information about blood flow $\dot{Q}$ at different anatomical or logical compartments to the mass transport information of the drug, as depicted in figure 3.15, while the blood flow itself can be described with different granularity by a CVS model [53].

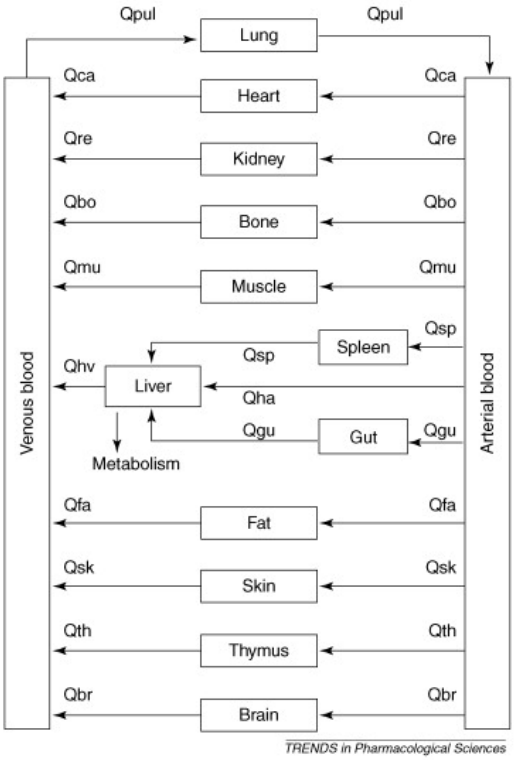


Figure 3.15: A typical Physiologically based PK model of the whole body. The tissues and organs are arranged anatomically and connected via the vascular system, with $\dot{Q}$ denoting blood flow [53].

But the clinical usage is not limited to the knowledge of drug concentration, one is interested in the time varying drug effect. Therefore, for clinical outcome, it is necessary to link the PK to the response of the patient, the drug effect and the science of the PD, which was also historically empirical characterized and based on population modeling approaches [2]. The basic strategy for PBPKPD modeling is to connect the knowledge about drug distribution at different compartments (e.g. given by figure 3.15) to the metabolism abstracted in the corresponding compartments. Usually the effect impacts the CVS and thus leads to a closed loop model.

Due to the heuristic modeling approaches on the one hand and the lack of informations on the other hand, the developed PBPK or PBPKPD [224] models are still very limited to experimental data evaluation or uncertainty in many parameters, which are aimed to be patient individual. One approach is here to simplify and use approximative description, e.g. to use simple mechanisms to explain complex chemical mixtures, due to the amount of uncertainty [68]. However such approaches are based on particular specific model and parameter fitting techniques and do not provide general validity, they were successfully used to describe toxicologic interactions of chemical mixtures [68][245].

To overcome the simple empirical approaches - which are by the way better than no modeling approach - in the last years a new worldwide process has started. The idea is based on bottom-up modeling from system-biology [110], which is actually also manifested in the purpose of the Physiome project [108][182]. Although, such approaches are in the very beginning and could not describe all mechanisms yet, there are already good descriptions for metabolism based PD modeling available. Especially the receptor-mediated solution is used to link metabolism mechanisms to the PKs [28], which point to the direction of a full virtual physiological human, including PBPKPD modeling for different levels of abstraction.

A corresponding link between metabolism level and PKs - although not automated yet - also exists, as presented e.g. by Ito et al. in 2005. Here, a methodology is presented to express the clearance from a three-step metabolism process by a 3-compartment model. The compartment parameters are related to higher oder compartmental systems, which e.g. express plasma concentration or liver activity [113].

Unfortunately PBPD modeling is very limited yet, especially due to the lack of human data. Many pharmacological data are based on animal studies [3]. The relation between body weight, surface area, organ or tissue weights, blood flow in different organs can be expressed with the different PBPD modeling approaches. But the model parameter - which are describing the above mentioned relations - are whether based on heuristics, or on populations, by the mean values for age, weight and other parameters. Therefore it is really difficult to use such models for individual patient predictions. 2005 Björkman et al. present a methods to predict such parameters e.g. also for children or infants, although they are very rare in literature [24].

3.4 Simulation Systems And Description Languages

The worldwide effort to have a fully described VPH can be backtracked to the International Union of Physiological Sciences (IUPS) PHYSIOME project [114], which was founded 1997[4]. The concept was to provide databases and integrative and analytical approaches for the study of physiology. Due to the complexity of biological systems, given by the vast amount of information available on different level of human physiology (genes, proteins, cells, tissues, organs), it was clear that for each level specialized mathematical modeling languages are necessary. Only by thins the relationship between structure and function at all levels of biological organization can be defined [108].

On this basis, for each level different markup or description languages have been created,

[4]The concept was already presented 1993

known from practical use or literature. CellML [38] [146] was created to describe the cell models, which was a first standard toward computational cell biology, providing standards for cell modeling based on XML and Mathematical Markup Language (MathML). In comparison TissueML and AnatML have been defined to model tissue and anatomical structures and finally PhysioML [35] was developed to provide a mathematical and functional organ systems and underlying processes.

The SBML [202][103] introduced 2003 was just a further development of the ideas already induced by the PhysioML schemes but more focusing on biochemical models and reactions, especially for simulation of metabolism and cell signaling. SBML is based on XML notation and can be regarded as the current existing ***standard*** for global exchange of various quantitative models for different languages and levels. Usually wrappers exist to transform the different standalone modeling scripts (e.g. CellML or Physiome MML) to the SBML standard [202].

By this, a general multidisciplinary framework is delivered, with the main purpose of integration of heterogeneous biological simulators into one global framework. This novel idea - unfortunately - is very hard to realize. Consulting the homepage of the organization SBML (SBML Software Guide/Matrix) [202] one can find 179 different registered modeling and simulation software products, which are applied in the field of biological modeling, however not all of them do support SBML notation. Most of the tools provide import or export functionality. But one is asking, what are the problems of SBML and over 179 different tools and why a new tool is necessary. To answer this, an overview of the most important tools regarding the topic of this thesis is presented in table 3.1.

Table 3.1: Table of important physiological modeling tools : One can distinguish between free (F), open source (O) and commercial (C) tools. Also the tools are compared regarding their abilities to model and simulated system of differential equations and compartments (ODE), stochastic systems (ODE) or dynamic stochastic models e.g. given by dynamic Bayesian Networks (DBN).

	F/O/C	Information	ODE	STO	DBN
JSim [182]	O	Java, multi-platform	+	-	-
BioUML	O	Java, multi-platform	+	-	-
QCP [9]	F	Windows	+	-	-
VCell	F	Multiplattform, Cell Analysis	+	+	-
BioNessie	F	Windows, Mac, Chemical reactions	+	-	-
BrainCirc [30]	O	Linux, Cerebral circulation	+	-	-
CellMC	F	Monte Carlo for Biochemical reactions	-	+	-
PK-Sim [81]	C	PBPK	+	-	-
SBMLToolbox	F	Matlab	+	+	-
BioRica [212]	F	Linux, Mac, Multi-scale modeling	+	+	-

All the tools, mentioned in table 3.1 are providing import or export capabilities for SBML language and many of them especially JSim [182] and QCP [9] provide many different models and even many tools are free or open source. All of them use compact model representation in XML language or similar compact scripts like MML applied for JSim.

One has to mention that the main contribution toward a unified description language for physiological phenomena was given by the NSR PHYSIOME [182] and the EURO Physiome projects [69], which are still very active and are further developing their tools and approaches.

While tools and approaches like JSim [182] are designed for general modeling and thus support different levels like cell and organ physiology, other tools like BrainCirc [30] focus on models for cerebral circulation.

Of course, one is also confronted with commercial aspects, for example PK-Sim [81] is commercial and not all tools are open source or provide full functionality for free. The QCP free software from the Universtiy of Missisipi Medical Center does not provide model manipulation at all and only the previous versions (QHP) allow the import of stand alone physiological models [186]. The latter is supporting a decoupling of model and model access and manipulation by interface definition. Full control is given to the user by a well defined XML definition, which was an inspiring pattern for this work [9] [186].

Some other main problems can be summarized as the following:

- The tools are not focusing on real-time applications. Most of the tools are based on Java (e.g. JSim [52] [182]) and concentrate on off-line simulation or web-based [239] access.
- Most of the tools are ***only*** supporting ODEs. The descriptions however differ due to the subjects, e.g. cell modeling (VCell) or circulation (BrainCirc).
- Stochastic simulation tools are very rare (BioRica, VCell). Usually the tools do not support the integration of both aspects, system of ODEs and stochastic models.
- None of the SBML capable tools support dynamic stochastic systems given by DBN.
- Despite BioRica, none of the other tools provide Multi-scale or hierarchical modeling capabilities. Therefore the models face critical complexity issues [104][202].

While many simulation approaches try to formulate their models as near as possible to real world aspects e.g. by mimicking physical behavior (physically based modeling), other approaches focus on simplifying model complexity issues by script based or knowledge based

modeling, especially because many tools and methods do not support hierarchical modeling and thus face serious complexity problems [104].

On the one hand, knowledge-based modeling approaches could not help to understand underlying physical phenomena, because they are only a summary of knowledge or observations. On the other hand, such approaches can easily contain aspects like nonlinearities or underlying stochastic phenomena, which are not used in simple system or physically based modeling. A good example for such a system is provided by the MVP [170], which is an agent-oriented simulation system especially created for medical teaching purpose. The virtual patient - grounded on knowledge-based models - is consisting of a physiological [117] and cognitive model-agent (double agent driven). Both models are grounded on ontological descriptions and the knowledge is represented in compact form by NLP [156].

Although these models are, traditionally seen, just "imitations" [46] and not real mathematical models, nevertheless they show that only mathematical models cannot fully cover all important aspects of the real physiology. One new contribution of the MVP is the integration of cognitive models (based on NLP), which are playing a significant role in patient physiological response as also proven by early research - not also irrelevant for cognition modeling of virtual electronical patients [153].

Whether stochastic methods are used, or know-ledge based NLP descriptions, which are very similar to logical chain networks, the author has shown that stoachstic causal networks, especially Bayesian networks as a generalization of HMM are very effective to model patient or human cognition and cognitive models [121] [120], therefore in this work, there will also be a focus on modern cognitive modeling approaches and their integration into the proposed system, discussed in detail in chapter 7 - "Mixtures of BN/DBN for Medical Simulations".

3.5 Static And Dynamic Bayesian Networks

Causal relations are very important for the description of diseases or physiological relationships. However, the relation between cause and outcome of a disease in medicine is usually very uncertain. To model this "*maybe relation*" a tool is necessary, that can handle uncertainties in information and influences and integrates them into one logical and usable inference system.

Here, Bayesian Networks (see also Chapter 2- Basics) come into play, because they offer an intuitive way to describe causal relations, which are affected by uncertainty. Until now highly complex models, found in literature (e.g. quantitative circulatory physiology (QCP), which

applies often more than 4000 variables to describe human physiology [186][9]) do not consider uncertainty in the parameters and assume fully described deterministic relationships. This is a severe limitation, which will be addressed by this thesis.

Of course, statistical information have been used a lot in this field so far, especially in the domain of PK PD modeling, but usually these models only apply simple population modeling approaches, which do not consider patient individual parameters [1]. One should not mistake BN with population models in this context.

Alternatively, other approaches, that simplify modeling complexity issues e.g. by script based logic and knowledge extracted by NLP and ontological semantics [170], do not consider the underlying physical models. Unfortunately, uncertainties in modeling usually are generally treated by regression analysis or population modeling such as the population pharmacokinetics (PopPK) [2] [211]. It is still a general guideline for the industry, although much patient-specific information is lost. Similar to knowledge extraction based on NLP, probabilistic decision networks (graphs) offer a way to represent information about physiology or human diseases. In contrary to ontological semantical chains, the chains in such networks - which are representing a generalization of Hidden Markov Chains - are describing stochastic processes. These techniques are robust against losses of representative knowledge in the modeling process and are including model uncertainties as well, which is very important for clinical data analysis.

Especially, BN are a very promising and useful instrument [148] to provide inference and classification for probabilistic reasoning [118] [71]. Dynamic Bayesian networks (DBN) are a generalization of Markov decision processes and extend the BN for the ***temporal modeling of dynamic systems***. For the discrete case, the DBN formalism is very similar to the case of static BN and thus the same approaches and algorithms can be applied in general, although the memory complexity may cause computational problems for huge networks [208].

By our approach, we address the domain of physiological modeling, showing the usability of DBN for modeling of complex physiological systems by hierarchical DBN sub-system modeling and the applicability for real-time physiological simulations for different purposes e.g. physiology simulators in medical education (see chapter 7 - "Mixtures of BN/DBN for Medical Simulations").

BNs, originally from the field of machine learning (where they massively become famous in the 90s [56]), are not only restricted to this field and are establishing as a common and useful tool in any domain, where knowledge under uncertainty is used for inference and analysis of complex systems.

The principal applicability of DBN for physiological modeling has been shown during the last decades in many applications [161] [247]. Additionally, DBN were analyzed according to their prognosis quality and temporal reasoning abilities, also known as Prognostic Bayesian Network (PBN) [237], e.g. for clinical patient management [230]. In many other areas e.g. system safety, dependability modeling [169][151], reliability analysis [27][20][136] and risk analysis [240] also for medical application [152].

BN/DBN are increasingly applied in medical and clinical domain, where tools are useful, which are capable to predict the outcome of diseases or disease treatments, models from supervised machine learning techniques are widely used, speaking of prognostic modeling. Here BN are applied to represent medical data in networks and allowing inference for prognostic objectives [175] [236] [237].

For medical application **diagnostic modeling**, by exploiting causal relations between variables from medical domain have been commonly used. Known as medical decision support systems (DSS), BN already find their way into the clinical daily use in the 1980s [242]. The formalism of BNs is easily to understand and therefore widely used for various decision support tasks, e.g. diagnostic reasoning, treatment selection, planning support and prognosis prediction [148][149]. In addition, due to automatic learning mechanisms, one is going to use BN also for understanding of data dependencies , e.g. in analysis of biological time series data [189].

Prognostic BNs which not only consider causality but also temporal nature of clinical data are of course very interesting for patient and disease management, because they allow predictions for future time slices [230].

DBNs offer various methods for time varying analysis under uncertainty [71] and actually all PBN mentioned in literature are nothing else than static or dynamic BN. In general DBNs are very capable for structural modeling based on multivariate time series data, which are very common in medical domain, as successfully demonstrated for diagnose of physiological conditions based on biometric multi-channel time series data from bio-sensors [145]. Fundamental research and analysis, including general methodologies for DBN theory is provided by Murphy [166], how is one of the famous founders in this domain.

Recently, clinical utilities have shown [236], that it is very effective to apply DBN(PBN) for modeling of time dependent relations in cardiac surgery treatments (integration in ProCarSur system), allowing a better planning and realization of pre-, operative and post-operative phases [237]. In 2009 PBNs have been used for prediction of pulmonary disease based on medical records and knowledge extracted by NLP [101]

3.5.1 Inference

The process of inference - also known as belief update - is defined by the actualization of information of network nodes in terms of probabilities according to known states of nodes (evidence nodes). For special cases the information for the evidence nodes is not strict, but soft or virtual, in case of BNs with uncertain evidential findings.[173]. Apart from such special cases, the update process can be done by various inference algorithms, classified in two main groups, ***exact*** algorithms and ***approximate*** algorithms.

The process of belief propagation can be expressed by combining messages from parents $\pi(x)$ and children $\lambda(x)$ of each node x to update the belief of this node $BEL(x)$ according to a certain evidence E. The combination is then given by $BEL(x) = P(X = x|E) = \alpha\lambda(x)\pi(x)$ and the messages will be updated iteratively [168].

The network structure affects the inference process substantially, because the complexity is directly related to it. Especially, single-connected graphical networks or trees provide better computational belief propagation attributes [118][244][50].

Therefore many approaches first try to simplify or change the network structure by equivalent transitions to achieve Junction Tree structure [118]. The removal of weak dependencies could lead to magnificent computational simplifications as well [126].

Usually, exact inference over time for DBN can be very complex and time consuming, because dependencies between different adjacent time slice lead to a exponential increasing of the space (memory) complexity [208]. Therefore, approximative inference techniques [29][168] like the factored frontier algorithm [167] are commonly applied. Modern techniques focus on transforming existing structure to a less complex one, which allows to use efficient algorithms, e.g. persistent DBNs (Singliar et. al 2008 [208]). A reduction of the complexity by removing weak dependencies can be achieved as well, however the transformation leads to approximate results [126].

3.5.2 Parameter Learning

The parameters of a BN/DBN - in other words the conditional probability distribution - can be adjusted by expert knowledge, which is typically applied for expert system design. But usually the networks can be very complex and map parameters from real life situations. Therefore, various algorithms exist, providing the mapping process from learning data sets to corresponding CPT or CPD for each node of the network, which is in general given by an

optimization problem.

During this learning process, one has to adjust the parameters (e.g. CPT) of the BN in such a way, that the CPD is sufficiently describing the statistical behavior of the learning data set. Usually the maximum likelihood (ML) is searched, which can be found by various Expectation Maximization (EM) algorithms [72]. One has to consider special cases if partially data are missing [58]. Additional information can be found in [166][161].

3.5.3 Structure Learning

The problem of learning graphical model structure is not limited to static or dynamic BNs [203]. Usually one distinguishes between learning from complete and incomplete data. The latter is consisting of samples with missing values, where it is more difficult to extract the correct model structure. One of the methods handling with missing values is the Bayesian structural EM algorithm by Firedman 1998 [72], which was improved 2000 [176]. The methods are based on optimization by the well known EM algorithm [58] including two steps: search over possible BN structures (expectation or e-step) and optimization of BN parameters (maximization or m-step).

BN not only allow to learn the structure based on real data. One is also capable to mix knowledge about parts of the network structure into this process - speaking of *structural constrains*. By this expert knowledge and real data relations can be used to model a complex system.

Generally, one needs to find the best (optimal) structure - DAG - according to a score function, that depends on the data and eventually on some restrictions [99]. This is task is shown to be a ***NP-hard problem*** [214].

Due to the complexity of BN structure learning, one is considering approximate algorithms, which can find close to optimum results, still in reasonable time. By applying various statistical tests or heuristic searches or by applying constrained-based approaches, recovering the structure on the basis of conditional independency tests, one can maximize the scoring function. Assuming infinite independent distributed data samples, the asymptotic limit (optimum) of different approaches is the same [214].

Two very useful and widely spread structural learning algorithms are the PC[5] and NPC. The PC algorithm [124] is dealing with structural constraints and applies conditional dependency

[5]Based on the work from Peter Sprites and Clark Glymour.

and independency tests to those pairs in the network, where no structure is specified. Due to the lack of PC to fully determine the directions of the dependencies in every case, normally the NPC[6] is preferred. Also here a skeleton based on statistical test for conditional independence is provided first.

One big problem for structural learning optimization problems is the space and time complexity. Many approaches, e.g. hill-climbing, dynamic programming and sampling variable ordering cannot handle many variables at the same time in memory (limited to 30 variables). Therefore 2009 a new branch and bound algorithm was presented which guarantees global optimality of the solution and was applicable to larger data sets (100 Variables) [55].

Due to the fact that the overall problem is an optimization problem - namely to find the optimal structure for given data - modern methods apply regularization techniques (e.g. L1 regularization paths) to force more efficient convergence [203]. Of course approximative algorithms exist, with the disadvantage that they can stuck in local optima.

3.5.4 Tools

A BN defines a directed acyclic graph over a set of variables (nodes), with a set of mutually exclusive states [118]. The qualitative part of the network is described by the connections (edges), which code logical dependency. The quantitative part is given by conditional probability functions [174]. Therefore the provided tools need to support both aspects, the qualitative and quantitative part.

Some tools like the BN toolbox for matlab [165], distribute the two aspects to other existing tools (e.g. for Bayesian analysis and network structure management). Especially the structure can be described by object oriented graphical network representation or be hold in matrices, representing existing arcs between the nodes of the network, which is ineffective for large networks, because many entries will be zero.

Various tools for Bayesian Network analysis are existing, many of them are based on Java (BayesianLab, BNJ). An overview of different software tools is also provided on Kevin Murphys website [165][128]. An extended extraction (current state of the art) of libraries which support C++ environment are given in table 3.2.

As one can see, most of the non-commercial tools do not support GUI for model manipulation. Hugin Expert [70], which provides a free limited version (Hugin Lite) is widely used, but is

[6] NPC stands for necessary path condition.

Table 3.2: An overview of important Bayesian Networks tools, supporting C++ development. One can distinguish between free (F), open source (O) and commercial (C) tools running on operation system platform Unix (U) or Windows (W). Different inference techniques are mainly supplied e.g. Junction Tree (JTree), Markov Chain Monte Carlo (MCMC) or others like the Lauritzen method.

	F,O,C	Platform	Info	GUI	Last update
PNL	F	W	JTree, C++ lib	-	2003
MSBN	F	W	Jtree, C# lib	+	2007
AgenaRisk	C	W/U	JTree, only API	+	2005
Hugin Expert [70]	C	W/U	Jtree, C++ API	+	2009
Bassist	O	U	MCMC	-	2000
Smile [66]	C/F	W/U	JTree, other C++ lib	+	2009

limited in open functionalities, e.g. open model description language [22].

The most actual and popular tool is given by ***Smile*** from the decision system laboratory [66], which provides a GUI support called Genie, the tool supports many different inference and learning techniques and the well documented library can easily be integrated into Unix or Windows environments and also well extended or combined with other libraries. Smile is also supporting DBN and corresponding methods [106], in contrary to many other libraries.

3.6 Simulation System Architectures

Due to the industrial background of medical simulation devices, literature on the field of system architecture are very rare. Many companies also hide the information about the way, they model physiology, due this is a new feature of their systems. The full scale simulator companies (Laerdal und Meti) also do not provide information about their simulator's internal architecture.

However, medical simulations, focusing on the physics and graphics simulation, provide various approaches for generic architectures e.g. as presented by CAML [51] or SOFA (open-source framework for medical simulations) [12].

The general functional anatomy of full-scale simulators was presented 1997 by Van Meurs et al. [232], which can be regarded as a *fundamental review paper* on this topic, also addressing the weaknesses of current systems and suggesting innovations. Here, the authors introduce the terminology "***engine***" of a simulator, which is the "component generating the physiological and pharmacological response of the simulated patient". Due to this context general designs are discussed as well as approaches to define control interfaces.

Generally, one distinguishes between script-controlled (e.g. SimMan Physiological trends [133]) and model-driven simulations (HPS physiological models [159]). However, the capability of script-based simulations, managing highly complex dynamic responses especially between interaction of different models (e.g. drugs, respiratory system, CVS) has to be questioned [232][59]. Although model-driven approaches are more extensive in designing appropriate sub-models for each tissue component, once they have been adjusted correctly, the time dependent dynamic interactions and responses are easily computed.

Still, a simulation instructor is needed to adjust the necessary model parameters for a specific case. To avoid this usually, educational specific script files are used to fulfill this task, speaking of script-controlled model-driven simulators [232]. However, due to rapid development of new drugs or new treatment techniques, the models need to be maintained and kept up to date. This problem was mentioned 1998 [231] and there was the suggestion to provide an "open model architecture" to integrate new pharmacological models into simulation systems, unfortunately the present simulators are still far a way from this idea. In an upcoming EU project 2010, the author in proposing to formally integrate physiological models into virtual avatar patients to provide an open interface for this problem.

The main construct, applied for human patients simulators in the model-based simulator design, can be extracted from literature [49][154][139] as depicted in fig. 3.16. In general, a multiple modeling method for computing systems is applied [154].

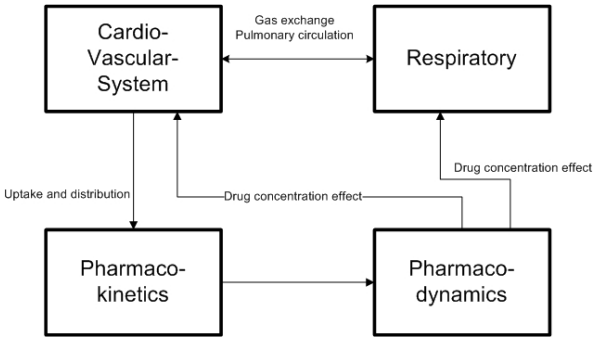


Figure 3.16: Model-driven simulator design: Main components of a human patient simulator are depicted, representing the view of the internal components of the physiological engine.

The connections between respiratory, CVS, PK and PD components are crucial for a realistic PBPKPD modeling. Usually multiple compartment models are applied for each of these components. A momentum transport model describes the flow changes in the CVS. According to the blood flow at each compartment, one can describe the changes in the PK by a mass transport model. In an interaction model (PD model), one is defining the effects according to the drug concentration at each compartment, here also effects to the resistances and capacitance

of the CVS model (see chapter 2 Basics) can be specified (e.g. for cardiovascular vasopressor Angiotensin II) [154], resulting in a closed loop model or circuit.

Additionally, for each component, four different levels of design were suggested: hardware, software, curriculum (parameter set) and exercise design [231].

Simulation of physiology is however, not only restricted to full-scale simulators or those for anesthesia education. Physiology is becoming an important underlying element of all medical simulation systems. A general composite architecture was proposed by Delingette et al. 2006 [57], underlining the importance of physiological models as a basis for simulator design, as depicted in fig. 3.17.

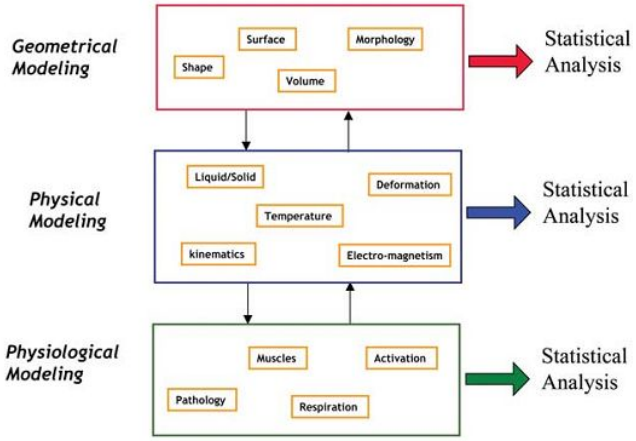


Figure 3.17: Simulator composition: Physiological simulation is the basis for simulator design [57], followed by physical and geometrical modelling.

Due to the many different input and output interfaces, which could be connected to todays simulators systems, the aspect of interface design and management is very important. The variety of the interfaces is given by *electro-mechanical*, *emulated sensors* and *alpha-numerical* ones.

A general approach for interface design, was given by Petty et al. 1999 to provide an high level architecture based on a runtime infrastructure, which defines the interfaces of the simulator HPS [179].

Unfortunately, although requested several times, especially by physicians, to provide a open-framework interface specification different manufacturers could share, this was never realized yet.

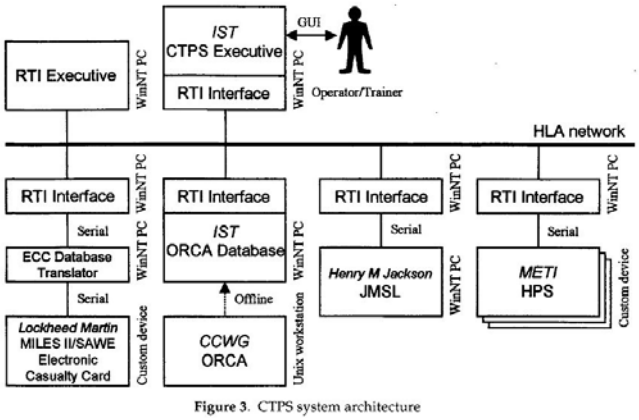


Figure 3.18: IO interfaces: A HLA (High Level Architecture) is used to describe the communication and interfaces between simulator (HPS), control interfaces. The Run-Time Infrastructure (RTI) is describing a set of services according to the interface specification. [179]

3.7 Summary

In the following, the relevance of the current work in comparison to the state of the art will be discussed briefly.

Medical simulator devices: Physiology was historically important for anesthesia simulators, where drug delivery and reactions are usually considered. Therefore only two main manufacturers provide physiological simulations and signals, both are full-scale simulators. However, nowadays physiology is becoming also important for part-scale, screen-based and fully virtual simulators, underlining the importance of this research area. The integration of physiology to such systems in comparison to current physiological simulator systems underlines several main problems:

1. Current system are not adaptive, flexible enough. Script based models are not realistic and physical-based models usually so complex that only specialized personal is capable to extend or manipulate them. This is a paradox to the daily growing number of new medicaments or therapeutic methods, which needs to be mapped by specialized models for such systems. Clearly, physical-based methods will provide advantages in realism, modeling and understanding of physiological systems, but only if the modeling language is intuitive, flexible and extendable. In this work, the author suggests therefore a XML based physiological model development for medical simulators, based on common standards, e.g. given by SBML. Only by such a generosity models can be exchanged and extended between the different branches, e.g. pharmacy, research and

commercial simulator companies. By this work the author demonstrates, that based on standardizes approaches compatible to European Physiome project [182], one is able to develop fundamental models for a physiology simulator.

2. Current systems fail to provide compatible and extensible hardware and software interface extensions. Due to a non generic architecture of current simulator systems, it is very difficult to combine the advantages of one hardware interface with another one, which is a contrast to the daily demand of different simulation centers. The hardware interfaces, necessary to physically emulate vital signals of an artificial patient, are dependent to the task the simulator has to fulfill. Logically an architecture is required, which supports easy plug-in of such devices according to the requirements. In this work, therefore - for the first time - the author does not only consider the "functional anatomy" [232] of simulator systems, but also focus on the architectural strategies and realizations. This includes insight views to hardware component design, and platform specific software development. Exemplary, during this novel work hardware emulation interfaces for simulation and emulation of vital signals for three clinical patient signals have been developed, which demonstrates the feasibility of the suggested architecture and the realism of the target simulator.

Physiological modeling:

Models of physiological processes are growing in number and complexity. In reaction to this fact, international organizations e.g. Physiome and SBML provide systematical and standardized methods and strategies. Databases and modeling languages have been developed during the last decades to support this process. However, current standards have still some limitations:

1. Although, the complexity of physiological systems is exponentially growing, especially because of fusion of different leveling models, e.g. cell and organ levels, hierarchical modeling is not supported yet by any of the tools, including the main references JSim [182] and QCP [186] and SBML [202]. The latter has planed to consider this aspect in future for the SBML level 3 standard. During this work hierarchical modeling realized by object oriented model based development was used for modeling of physic-based medical physiological processes. The benefit, herewith, relies on easier, clearer and more comfortable design and manipulation of system components during and after the model development process.

2. Most of the physiological model development and simulation methods only provide limited functionality. Combined probabilistic and deterministic modeling approaches are

very rare. The important comparative languages JSim and QCP only provide systems of first order differential equations. DBN are not supported in general for this purpose, although they offer an easy and understandable way to model dynamic physiological processes under uncertainty based on learning from real patient data. DBN are especially useful, if only data are available and underlying models are unknown. In addition medical expert knowledge can be used to influence the model development process. Therefore in this work, to provide maximal flexibility in system description and model development, a hybrid approach is used, which supports both deterministic modeling, based on algebraic and differential equations and probabilistic modeling based on dynamic Bayesian networks. Additionally our system is extendable and provides the possibility to integrate other modeling frameworks, if necessary.

3. The languages used to model physiological processes form an inscrutable conglomerate. Each language is facing advantages and disadvantages comparing to others. While MML from JSim is not based on XML and thus is less hierarchical or object oriented on the one hand, on the other hand it provides a more compact mathematical representation. QCP in contrary is fully using XML model notations, which is again a standalone language independent from SBML. The latter, which has defined a world-wide standard is designed very general, such that a model developer has to pass more than 10 different XML layers (tags) to be able to define a function variable. Although a general standard as SBML is necessary to guarantee international uniform model exchange, most physiological modeling tool are using their own specific language and provide model wrappers. In general DBN are not supported by SBML yet. We suggest a specialized XML notation in our system for model development and description, similar to the QCP XML languages. Our notation is extended to enable hierarchical modeling (composition) and stochastic processes by including standard BN and DBN models and provided variable interfaces.

BN/DBN mixture for medical simulations: In comparison to dynamic Bayesian networks, static BN have a long history and are established successfully in different areas and applications. One reason for this is the space and time complexity, which leads to huge computational processes, making DBN inapplicable for standard applications. However, due to optimization of typical DBN algorithms and methods during the last years, probabilistic inference is increasingly influencing many research areas.

In this work BN and DBN mixtures have been considered in detail for medical simulations. The impact relies in the fields of cognitive simulations as well as physiological simulations, which are indeed strongly connected.

In both fields, models and simulation up to now are based on regression analysis, Structural Equation Modeling (SEM) or logical chains [120]. Applying BN and DBN in contrary to other methods, provides a better fit to original data and includes the logical dependency between the different parameters in form of a graphical representation.

4 System Architecture

In the following chapter the architecture of the proposed system will be presented in detail. First of all, there will be an overview of the system and its components. Then, the nucleus of the simulator system, the simulator engine - which is basically realized in software - will be presented. It contains essential methods to solve complex mathematical models, deterministic as well as probabilistic onse.

The model management strategies, which have been improved in comparison to existing software, will be explained afterwards, containing modern object oriented methods and strategies.

Another important aspect is the model view and manipulation, which is essentially important for hardware and software interfaces. Well founded strategies and methods to provide smart solutions for this aim will be presented next. At the end of this chapter, a generic hardware architecture with the purpose of designing adaptable simulator hardware interfaces will be discussed.

4.1 System Overview

In figure 4.1, the overall system architecture is shown. The main parts of the system are listed in the following. Their explicitly explanation is the major purpose of this chapter.

- Model Loader
- Model Solver
- Simulation Kernel
- Model View Controler
- Hardware Interfaces

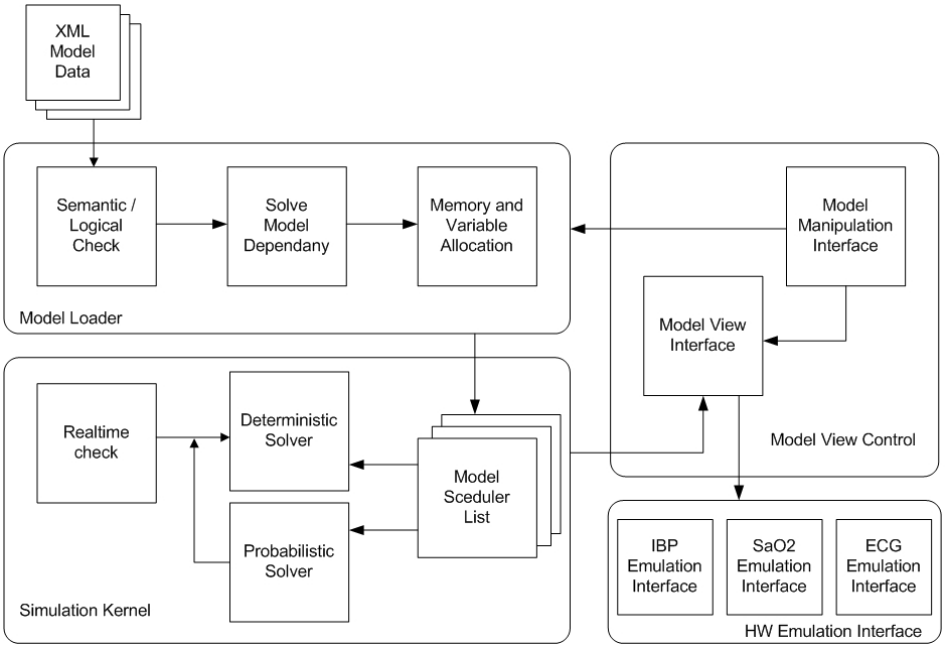


Figure 4.1: Overall system architecture

The basic strategy is to encapsulate logical models (e.g. XML models), physical models (model object in memory), the model processing (simulation kernel) and input/output processes (visualization, hardware interfaces, model manipulation). Encapsulation as a useful and underlying principle is the basic of any simulation environment and can thus repeatedly be found with small differences in other systems for medical simulation, e.g. Sofa [12], CAML [51] or virtual reality and game engine development [22].

The main advantages are flexibility and independency. For instance logical models are independent from the platform the software kernel is implemented on. They provide a flexible way to extend functionalities, which could be realized later inside the kernel. The kernel itself is also encapsulated and can be realized in different ways if one think of platform specific realization strategies e.g. for GPU accelerated applications. Referring to JSim and QCP additional flexibility is provided by including front-end model view and manipulation mechanisms inside the logical model description (XML). By this, the front-end user is capable to design user defined input and output interfaces.

Physiological models are described by XML files, which are processed by the **Model Loader**. Here, after applying semantic and logical checks on the model description, model dependencies are solved, which is one main advantage of our system. In this step, model parameter referencing and/or dereferencing is controlled as well as breaking cyclic dependencies into a serialization streams.

After these steps, a model and its parameters are allocated into a separated model environment, which could have different character, e.g. differential model, algebraic model or a Bayesian network model or any combination. Every access to a model instance or its variables is managed by a model controller, applying principles of Model View Control (MVC), such that model manipulation and model view is encapsulated from the model itself.

The **Simulation Kernel** is consisting of deterministic and probabilistic solvers, which are processing models in schedule each simulation step. If the time needed for processing the data exceeds the duration of a time slot that is available for a simulation step, the real-time condition is not satisfied. Therefore the kernel process, which could run as an autonomous thread, has the highest priority. The realization of the solver mechanisms is encapsulated again from the architecture. By this, one can for instance use exact or approximate belief update for BN solver processes. The main advantage of the proposed simulation kernel is, that it supports mixtures of systems of ODE and static or dynamic BN. By this hybrid approach, we can use the benefits of both approaches, which leads to very realistic patient simulations. Those components of a system, which are perfectly described by Linear Time Invariant (LTI) systems, e.g. the circulatory system, can be described by a system of ODE. Other components, where current methods have difficultly to map the system behavior to deterministic and parametric model, can be described by learning probabilistic networks from real data.

The **Model View Control** block is responsible for model view and model manipulation. For physiological simulations one is essentially interested to induce special events, e.g. by activation of drug delivery or special symptoms or diseases and changing model parameters. The model manipulation interface provides functionalities to change variables, that are supported by many other simulation systems too, but in contrary to other systems like JSim [182] and QCP [186] parameters can be modified during a simulation run. The novelty of the proposed manipulation interface is to change the simulation system itself, not only by parameter changes, but also by model changes. This goal is realizable, because hierarchical modeling is supported by our architecture, in contrary to other system. Thus, the user can load and unload components (sub systems) of an overall simulation system during the runtime. This functionality enables the possibility to load and activate physiological defects to the system, e.g. by including new compartments in case of shunt simulation, as well as other events like medications or intervention by loading corresponding models.

Physiological simulation engines are used in three ways, whether screen based, as part a full-/part-scale simulator or as a combination of the others. We encapsulate this view in the architectural design of our simulation environment, such that a general system, independent from the realization purpose, is delivered. Logically, the final realization should be defined

by the output interface of the system, which is in our architecture managed by a model view manager. Here the system can adaptively include only screen or computer based views and interfaces for internal physiological signals, as well as for external visible ones (e.g. ECG or blood pressure). Additionally, it supports external emulation interfaces, which are usually related to peripheral hardware.

Special purpose emulation hardware is used to emulate vital function parameters e.g. 12-channel electrocardiogram (ECG), spatial oxygen saturation (SpO2) and invasive blood pressure (IBP), in addition to virtual software monitoring. By this approach we provide virtual and real patient monitoring [6] for the ICU application. In the following, the particular components of the system (compare figure 4.1) will be presented in more detail.

4.2 Simulation Kernel

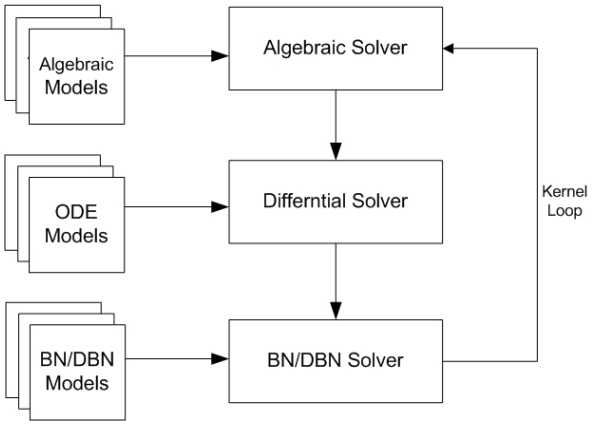


Figure 4.2: Main kernel loop. Three particular solvers are addressed in each simulation time step. By this mixtures of algebraic and differential equitation models and static and dynamic Bayesian networks models can be proccessed.

The simulation kernel is the engine and main component of every simulator environment. Usually within this component a computational loop - the kernel loop - is realized, as depicted in figure 4.2. This loop is often computational intensive and needs to be controlled according to timing constrains. For our system, the main goal is to solve different models for a given time step within the sampling rate, which is adequate for a **real-time** simulation.

The sampling rate is limited by the system complexity on the one hand and by the visualization and emulation on the other hand. While for haptic critical emulations, sample rates of 1 KHz are common and thus special so called haptic loops are used in medical simulations, for our physiological simulator the emulation hardware is sampled in the region of approximately

10-50 Hz, which is thus limiting the overall system timing t_{max}.

Nevertheless, for each cycle of the kernel loop, the computational time t_{loop} needed should fulfill the limitation $t_{loop} \leq t_{max}$. This is a so called soft real-time condition, which needs to be fulfilled for each simulation cycle. Certainly, this condition is very dependent on the realization platform (essentially the computational power) and the complexity of the *loaded* physiological models. The real-time block (figure 4.1) supervises this condition for each kernel loop. In special situations, for example during load of a very complex model, this condition may be unfulfilled, because both processes, the model loading an the model solving, may take more time, than allowed per definition by the simulation sample time.

For a real-time critical system, this fact is unacceptable, but for simulation purpose and under soft real-time condition acceptable, especially because event based smaller models, that have been used in praxis, are loaded usually to the system during run-time and for them the real-time condition is fulfilled.

In general, this problem can be addressed by a real-time scheduling, where new system components are loaded in real-time and in schedules into the runtime environment but activated/solved only, when the model is loaded completely. However, there will be always a model initialization procedures, which is time dependent to the actual time step and thus individable and limiting the real-time requirement.

During each simulation loop the computational process is delegated to the solver components (compare figure 4.2), including solvers for algebraic expressions, differential equations and DBN and BN simulation, which will be explained in the following in more detail.

4.2.1 Algebraic Solver

Algebraic expressions and functions are realized by a mathematical parser interface class «ParserInterface». Each model has its own parser interface, which supports **inter-model communication**. This is basically important for hierarchical modeling, such that different models can exchange variables.

Particular variables of each model are binded to the mathematical parsers by a definition *defineVariable(string name, double& memory)*. Any manipulation applied on a variable will thus be treated directly on the memory allocated for this variable. This leads to a very fast and direct access to each variable value, even for distant models.

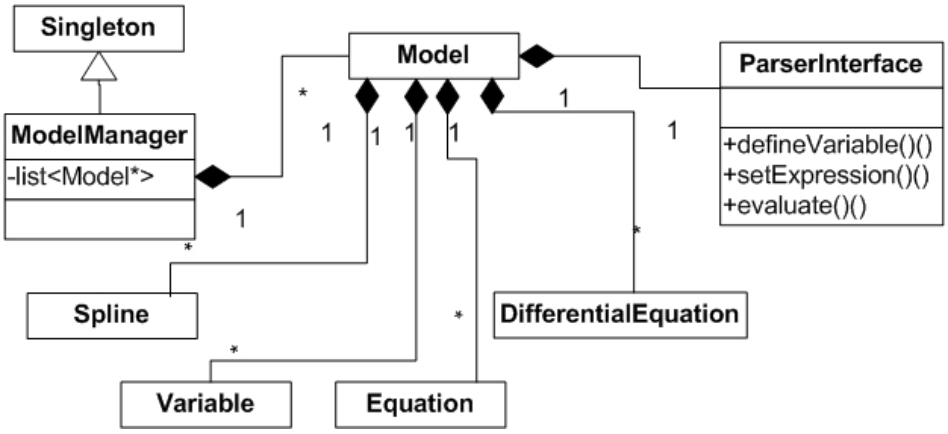


Figure 4.3: Relation between «ModelManager» and «ParserInterface» objects. Usually particular «Model» objects are loaded and thus managed by the «ModelManager». Each model has one «ParserInterface», which is acting as an interface for variable exchange between different functions of a model or between many different models (inter model communication).

Algebraic expression are defined by using the method *setExpression(string expression)*. The result of the expression is evaluated by using the method *evaluate()*, which is important for the kernel loop. Different exception handling routines are supported to avoid system crashes during evaluation of expressions, e.g. due to incorrect semantic user inputs.

While predefined functions can be used by the "function()" call within the expression definition (e.g. "x = sin(y)"), splines can be evaluated by the "spline[]" call (e.g. "x = spline[y]"). Additional spline methods and user defined functions can be easily included into the system, if necessary, which is one of the advantages of the proposed architecture.

The principal composition is demonstrated in figure 4.3. As one can see the «ModelManager» class consist of one or many model objects, which can be composed of many variables, functions, splines, differential equations and one specific mathematical parser interface. The benefit of this architecture is flexibility and extendability, which is the major lack of other systems. By this way inter-model communication is supported, which is crucial for hierarchical modeling.

4.2.2 Differential Solver

The differential solver has to solve a system of differential equations, while the particular equations could be distributed in different models. Therefore, the solver class «ODEManager» is a singleton, which can be accessed from separated models (compare figure 4.4). As the basis

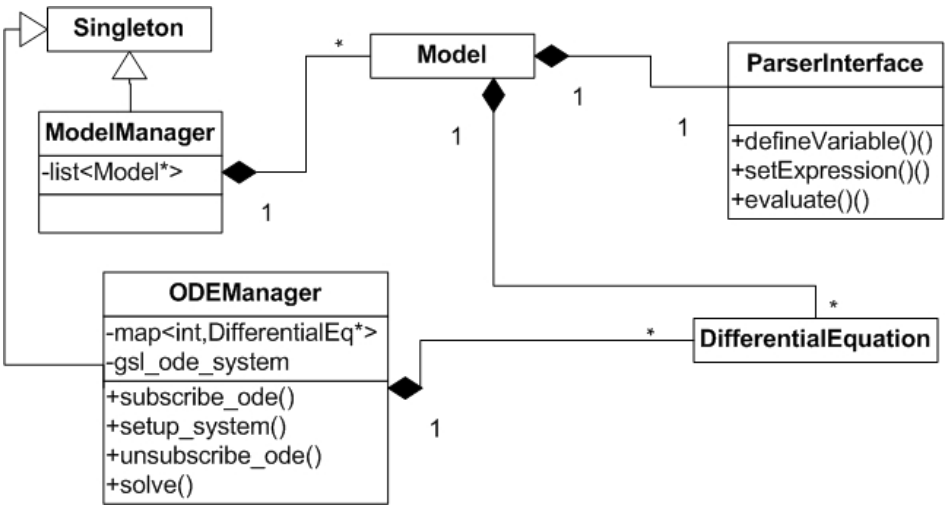


Figure 4.4: ODE management. The «Model» class is composed by «DifferentialEquation» class objects, where differential equations are specified formally. The system of differential equations is managed by the «ODEManager» class, where the numerical solutions of the ODE system are calculated. Access to parameter and functions is granted by the «ParserInterface» class. Both manager classes, «ModelManager» and «ODEManager» are using singleton pattern by inheritance.

for this manager the GSL ODE solver classes and methods are applied [87]. Because the GSL is originally written in C, the function callbacks have to be redirected by a wrapper function class to the corresponding functions distributed in different C++ models.

The novelty relies on the adaptivity of this manager because the ODE solver and manager class keep the systems of ODEs dynamically, in other words extensible. Which means, that the ODE system dimension given by the Jacobian matrix (compare equation 2.28) can change during the runtime.

Every time a new differential equation is loaded to the system the «ODEManager» map will be updated, using the "subscribe_ode()" method, actualizing the corresponding ODE system, derivative functions and Jacobian. If the differential equation is removed from the system, the "unsubscribe_ode()" method is applied. Each differential equation has its own unique identity, which is corresponding to the position in the system state vector.

During initialization or update of the system of ODEs the temporal initial states have to be provided for new differential equations and will be stored for known ones. The differential system will be updated and setup with the "setup_system()" method. All this elementary tasks are delegated to the «ODEManager» class. The real value of the derivative functions is calculated by using the algebraic solver interface, supported by the «ParserInterface» class.

4.2.3 BN DBN Solver

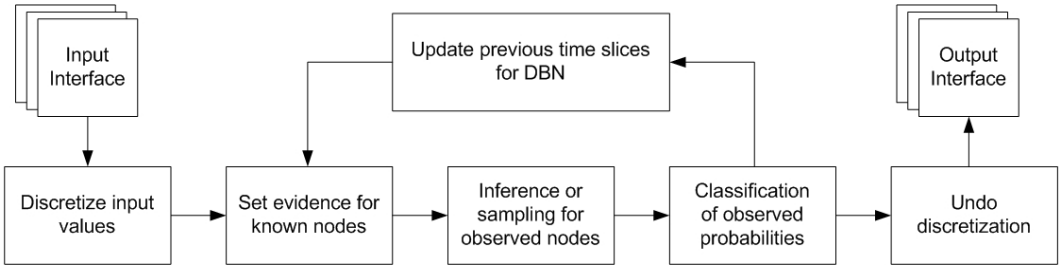


Figure 4.5: BN/DBN solver process cycle

Integrative physiological modeling describes the human body, consisting of many different sub-models with different time granularities. To cope with this higher complexity, we apply a hierarchical object-oriented strategy with sub-models as building blocks to model the corresponding physiological sub-systems. Every sub-system can also be described by BN and DBN with anchor nodes to be connected to other models.

By this, sub-models may be added or removed dynamically from the simulation environment containing probabilistic modeling approaches as well, such that complications, interventions, and medications, as well as other types of events (which are affecting the human physiological system) can be described by static and dynamic Bayesian networks.

We consider a hierarchical decomposition of systems into corresponding sub-systems being modeled autonomously. Each of these sub-systems is characterized by predefined attributes and parameters, being stored in XML format and being read at system startup. One important parameter is the size of time-steps represented by the time granularity of the network [17]. Especially, for dynamic BN the sampling time between the discrete time slices will play an important role for the simulation, as will be presented in chapter 7 - *Mixtures Of BN/DBN For Medical Simulations*.

For the simulation kernel this sampling has to be an integer multiple of the main simulation sample time. The BN/DBN simulation process is thus executed with this sampling frequency and is essentially defined by the schedule depicted in figure 4.5. The different process components are described in the following:

- **Input and Output**: Since evidence and inference are the main determining processes in a BN, we use a managed list of input (evidence) and output objects (marked for observation).

- **Discretization and Undo Discretization**: Input and Output objects can have differ-

ent types and are defined mathematically in their own signal space. Considering discrete BN, continuous variables have to be transformed to their discrete state representation and vise versa.

- **Evidence and Inference**: Once evidence was applied to the BN by the knowledge of input variables, the Bayesian inference can be applied to calculate - approximative or exact - the probabilities for the output variables.

- **Classification**: These probabilities needs to be mapped to the knowledge of the state of the variable, which is called Bayesian classification and can be done by different methods, e.g. maximum likelihood classification or sampling. The interested reader can find additional information about optimal classification of continuous data in literature [74][73].

- **Smoothing**: For a dynamic BN the temporal information from each time slice has to be used for the next time slice. This process - also known as smoothing - is in principle a temporal evidence which has to be applied for all dynamic nodes of a network.

To control the above mentioned processes the structure represented in figure 4.6 is applied. The manager pattern is also applied for this purpose. The «BNInterface» class is delegating the input(«EvidenceInterface») and output(«ObserverInterface») of the network, where evidence can be set to the network (set_evidence()) after discretization (discretize()) and classification of the output variables is applied (classifiy()).

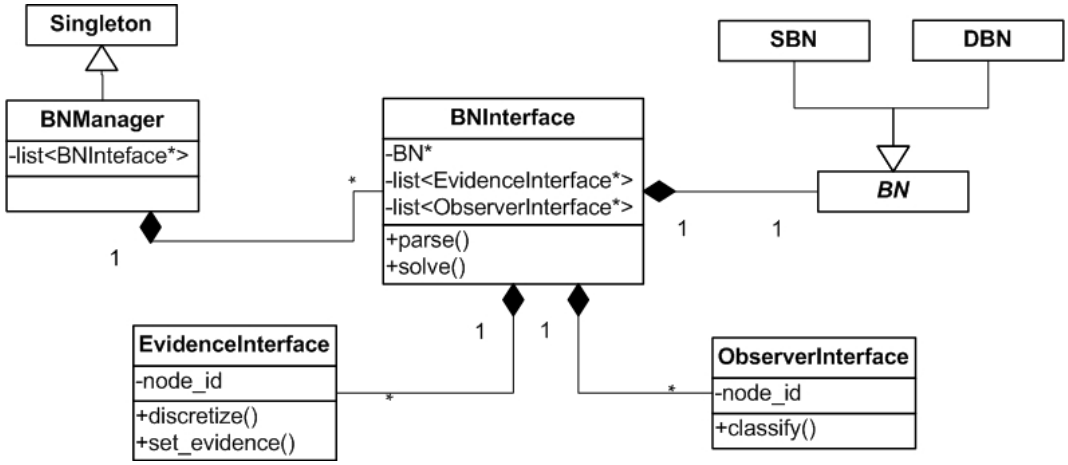


Figure 4.6: BN/DBN Management : static and dynamic BN are inherited realizations of the abstract class BN. I/O is managed by «EvidenceInterface» and «ObserverInterface» class objects, composed in a «BNInterface» class object.

4.3 Object-Oriented Model Management

4.3.1 Modeling Scheme

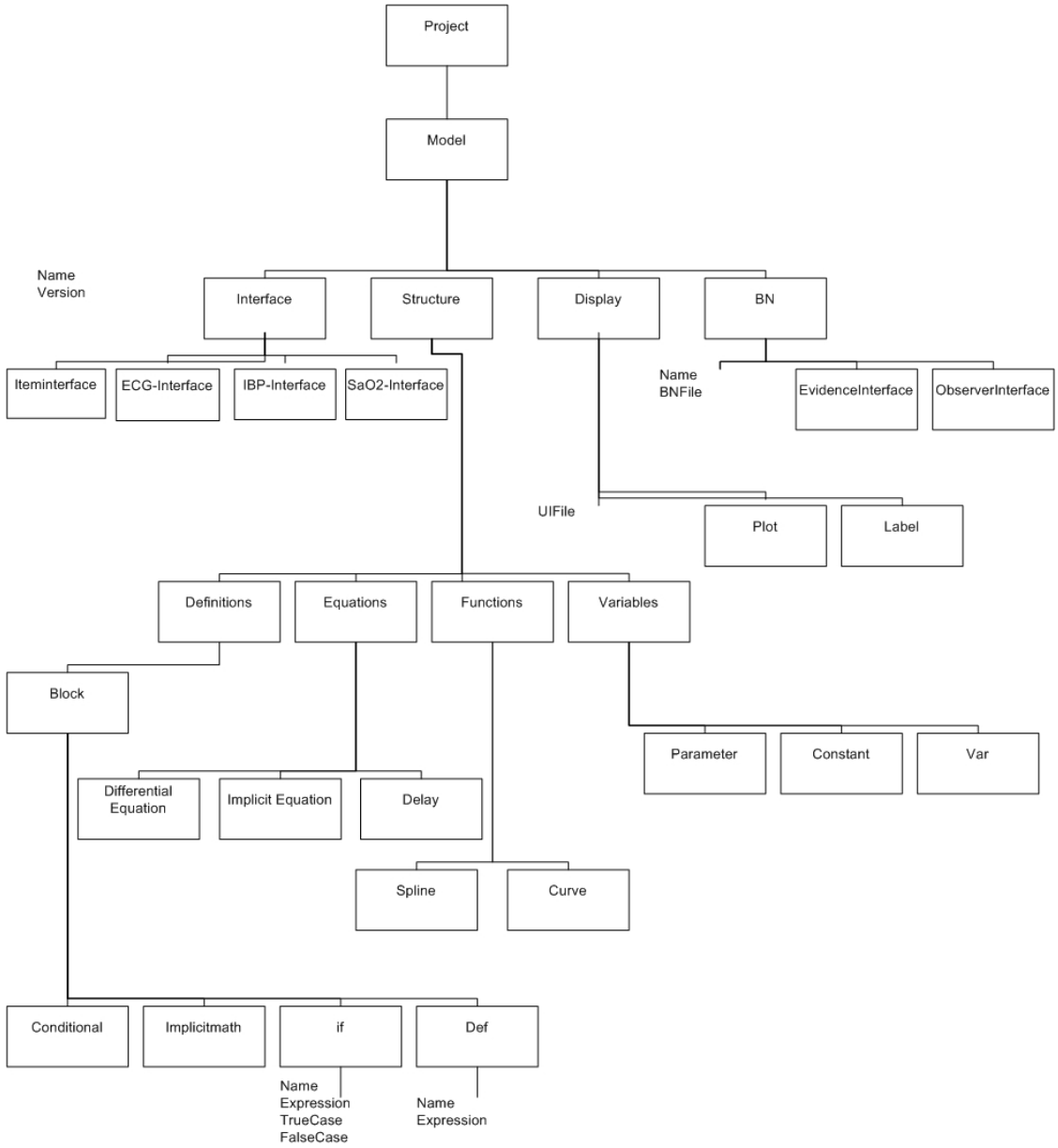


Figure 4.7: XML modeling tree: A hierarchical XML tag structure is used to describe interfaces, mathematical structures, IO and BN components.

The modeling is managed by XML file notation, because it offers flexibility for further extension and manipulation. The overall modeling tree is depicted in figure 4.7.

One main goal of this system is to provide a better modeling tool for the description of known physiological phenomena. One big advantage is given by allowing hierarchical modeling, which is not supported by other tools like Physiome or QCP. With other words, to allow models to be composed by others. As one can see in figure 4.7 each project may consist of one or many models. To hierarchical access parameters of other models the "->" Notation was included into the parsing process. So in general, when the notation *model* is used, this could mean as well a higher level as a lower level (sub)model.

Considering the XML modeling scheme, one can see that each model is consisting of four different descriptions: The <Interface> tag is used to describe hardware and software interfaces, used for emulation of signals. This declaration is also necessary, because by this the hardware interfaces can dynamically be binded to different signal sources. The <Structure> definition is used describe mathematical relations of any kind as logical, explicit or implicit equations, functions, splines, look-up-tables or differential equations. With other words every detail, that is necessary to describe the deterministic part of a model.<BN> is thus used to describe the non-deterministic part of a model, by integration of BN or DBN classification and sampling into the model.

The <Display> definition is used to provide an easy and user defined way for GUI development, e.g. to visualize certain parameter or manipulate them. For this purpose the QT user-interface designer can be used to generate a personalized (user specific) view and content (*.ui), that can be loaded to the simulation environment dynamically.

4.3.2 Model Management

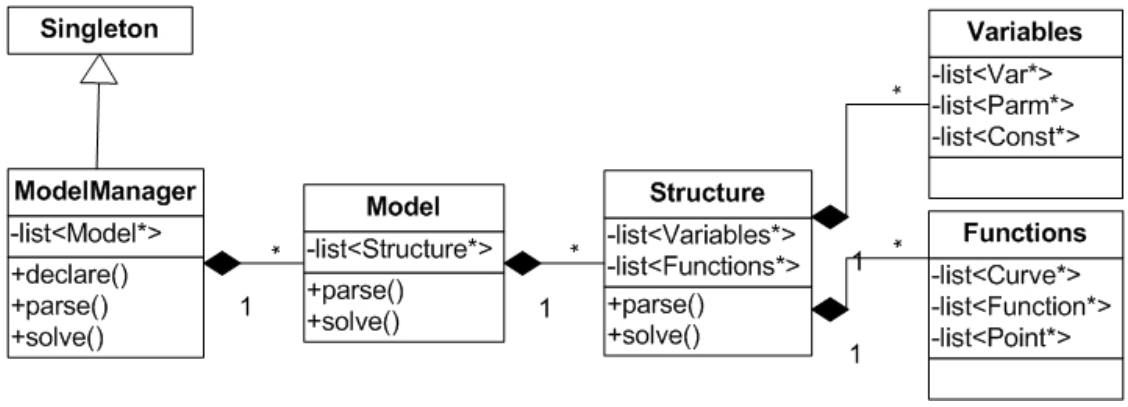


Figure 4.8: Hierarchical model management: Model building and solving process is delegated hierarchically to subclasses. This leads to an easy extensible and maintainable structure.

The «Manager» [1] and «MVC» pattern are very useful to initialize and control non static environments. By this, maximum adaptivity during runtime can be ensured on the one hand and on the other hand the physical and logical models are encapsulated. The principles have been applied, whenever necessary, referring to the style managers are used in game based development. This means to encapsulate independent process loops by independent managers and limit the data exchange between them to a minimum, as shown before in figure 4.2 for the main kernel loop.

Additionally, hierarchical composition and management is also applied for the model management itself. This is exemplary visualized in figure 4.8. As one can see, composition and inheritance - the two main OOP principles - are applied in this design. Herewith, the class structure is equivalent to the modeling scheme of the XML notations (presented in figure 4.7). This leads to an easy extensible and maintainable code and design.

Therefore, for the mathematical solvers, different manager classes have been used («AlgebraicSolver», «ODEManager», and «DBNManager») as well as for the management of the overall models («ModelManager») and the control of the views and interfaces («InterfaceManager»).

4.3.3 User Interface And Visualization

Whenever one is facing the task of designing a user interface, the Model-View-Control paradigm, one of the most important object oriented design pattern [77] and strategies, should come into mind. Although the idea was born in 1998 [129], and has been extended and further developed (e.g. the Model View Presenter (MVP) pattern [82]), the methodology and principle have not changed substantially ever since. The issues are to provide *reusability* and *pluggability* [109]. In other words, to provide flexibility and modularity for the system interfaces and model manipulations [115].

This is addressed by a logical separation of knowledge (models), views (interfaces) and manipulators (controllers). Interaction between these three components are realized by communication mechanisms [235][77].

By this way, it is very easy to provide multiple user interfaces [10] for one model, representing knowledge in different ways. This is important for the presented system, especially if one thinks of several views and controllers, which can be added to one model.

[1] The manager pattern is often used to generalize operations on objects from the same interface.

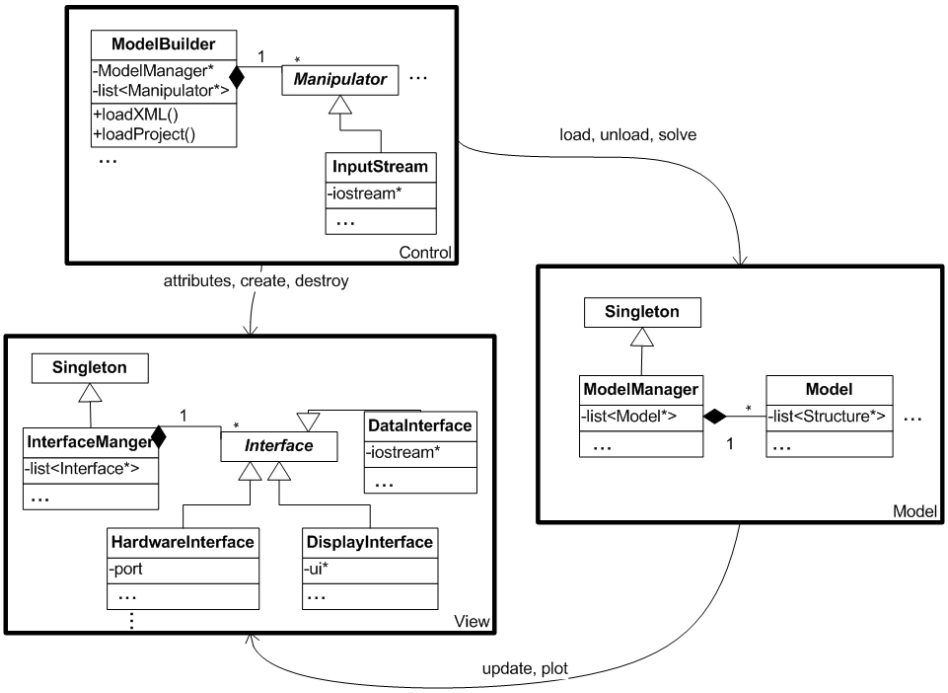


Figure 4.9: The Model View Control paradigm is applied for the system architecture.

Of course the MVC strategy has been realized on many different platforms and in many different ways for C++ programming. The platform independent QT development library [188] nevertheless, due its widely usage for the development of GUI programming, has fundamental advantages in realizing the MVC architecture. QT simplifies also the communication of separated logical levels by a signal and slot mechanisms.

Essentially, different manager classes have been taken to delegate the particular function of the MVC paradigm as depicted in figure 4.9 in simplified form. The "model" part is represented by the «ModelManager» and «Model» classes and the hierarchical composition of Model structure, discussed before. The "view" part is organized by an «InterfaceManager» class and by the composition of different interface objects. Hereby the «Interface» class is an abstract base class for derived classes like «HardwareInerface» for emulation devices, «DataInterface» for data recording and «DisplayInterface» for visualization. The "Control" part is given by the «ModelBuilder» manager class, which is responsible for model manipulation and other manipulation interfaces, e.g. for parameter changes. The specific commands are handled by standard template library (STL) iostreams, such that they can easily be transported over network for remote control processes. Thus, «Manipulator» is an abstract base class, which can have different realizations in derived classes.

4.4 Hardware Interface Components

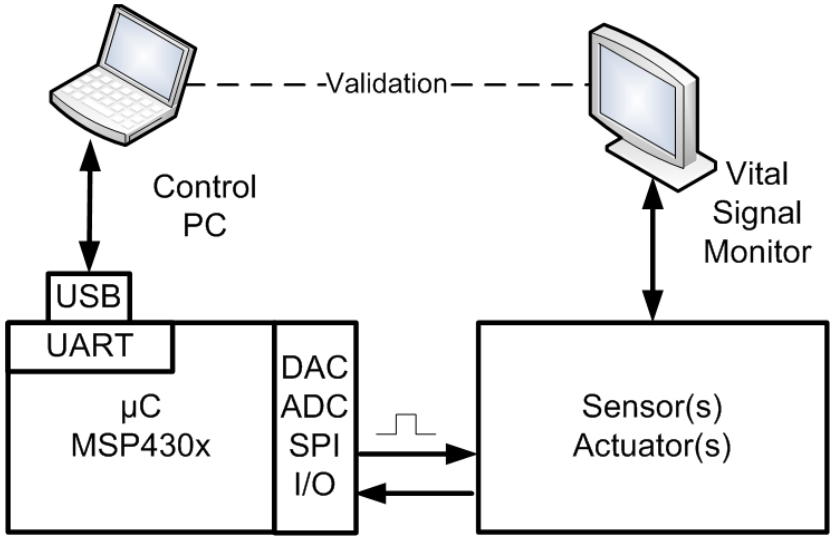


Figure 4.10: Generic client server architecture for physiological emulation hardware interface development.

For the hardware interface component development a generic architecture, based on a client server model is suggested, as depicted in figure 4.10. The server is a **PC/computer** system, while the client is a **micro-controller** (μC) with IO functionality. Universal Serial Bus (USB) is used as a basis for data exchange between client and server components of the simulator. This decision relies on successful attempts to use bus systems for simulation application [179] [37]. The USB standard is well developed and established for data transfer and is supported by most of the standard computers. Therefore the system will be applicable to different computer platforms. The data rates 12 - 480 $Mbit/s$ (USB 1.0 and 2.0 specification) are fully satisfying the client server communication task.

The generic architecture, for the first time presented by the knowledge of the author for this problem statement, is applied and realized for three different hardware emulator interfaces, which can be connected to the software simulator server. By this the feasibility of the suggested architecture is shown.

The clients are realized as embedded systems with integrated μC and the attempt is to force as much calculation power to the servers themselves as possible to reduce the bus traffic to primitive command and data exchange. The MSP430x series have been used as μC development environment [112], especially because it is one of the less μC platforms, supporting JTAG and advanced runtime debug modalities (see also figure 12.1 in the appendix).

The μC is used to measure **sensor(s)** information and to control **actuator(s)** by I/O interfaces (e.g. ADC, DAC and SPI). The result of the physiological signal emulation is measured by a **vital signal monitor**. And data comparison between measured signals and model values (originally calculated on the server) can be used for optimization during a **validation** process.

The benefits of this design can be summarized as the following:

- By the client server architecture, computational expensive calculations can be outsourced from the external hardware modules to the more powerful computer/pc server.
- By the USB interface to the PC plug&play functionalities can be used.
- The ultra low-power μC enables stand-alone applications, supplied by the USB 5V power, as realized for ECG and SaO2 emulation interfaces exemplary.
- Fast on board hardware processing can be applied for time-critical sensor application, e.g. SaO2 emulation with 1-10 KHz sampling rate.

4.5 Summary

The presented architecture unite the advantages of many different systems into one and overcome current state of the art problems and limitations. The resulting benefits and attributes can be summarized as the following:

- By this system, well developed physiological models can be integrated into a medical teaching simulator environment for the first time.
- The architecture is composed in a way, that the system can be used for both, screen based physiological modeling and simulation, and for medical teaching simulation using hardware emulation devices. This mechanisms are managed by a model view delegation.
- The system allows the combination of different modeling mechanisms, by integration of algebraic, differential and BN/DBN solver classes into the main kernel loop. Therefore hybrid simulations with systems of ODEs and static/dynamic Bayesian networks are possible and can be used to drive emulation hardware components. This composition is unique and offers much better modeling possibilities.

- Models are composed hierarchically in XML, the same way they are managed by a model manager inside the software architecture. This innovative feature is missing in other environments yet, as well for screen-based modeling as for full/part scale simulator modeling. It provides on the one hand more flexibility for design and development of complex models and on the other hand a better overview of the system composition.

- By applying dynamic probabilistic causal networks, it is possible to simulate systems, based on learning from real data samples and expert knowledge. The modeling becomes more intuitive, because it describes causal relations and influences. This aspect is very innovative for medical simulator applications.

5 Realization

5.1 Software

The software has been realized by applying various modern programming techniques in C++. If one is aiming a fast and real-time software, which can target different platforms and operation systems (platform independent), support object-oriented programming and provide well-developed libraries for scientific programming C++ is a good choise.

The necessary source Lines Of Code (LOC) to realize this project are shown in table 5.1. The most effort has been investigated in the realization of the PhysioSim real-time simulation environment. Additionally, C++ programming was applied for the implementation of the simulator hardware μC and for communication with patient monitoring devices.

Name	LOC	Files	Comments
Remote monitoring interface	12.605	65	17%
PhysioSim simulation environment	49.253	316	18%
MSP430 μC for clients	3.308	33	45%

Table 5.1: Overview of Lines Of Code (LOC), files, and documentation (DoxyGen comments), needed to realize different parts of the project. Only *.cpp and *.h C++ sources are considered.

5.1.1 Design Pattern

Design pattern are very important to provide modularity of the source code [137]. Therefore, during the realization of the source-code various design pattern have been used, e.g. "Singleton", "Manager", "Factory" , "Controller" and "MVC" among others. "Dynamic binding" is essentially used to realize function callbacks and remote variable access, which are important for event-based applications.

5.1.2 Multi Threading

Threads and multi threaded programming is one important aspect of the software. Due to the fact that the solver kernel of the simulator has to deal with model dependencies, it is difficult to extract reentrant or thread safe functions. Therefore there is a septation between kernel solver thread which acts as a producer and a thread pool for user interface threads, which act es consumers. Essentially the kernel solver thread has the highest priority in comparison to consumer threads, which e.g. update visualization interfaces. For realization of this approach QT's core module classes <QThread> and <QThreadPool> have been used [187].

5.1.3 Server functionality

By the MVC mechanism applied for model manipulation and model view, one can address the simulation system to load and unload models during the runtime, change model parameters in the hierarchical model structure and ask for new views (interfaces) for particular parameters. Therefore the system is independent from the target usage.

By this, a basis is delivered to use this system as a simulation server, where other simulation clients can dock on, to change model parameters or asks for current calculated vital parameter. One can think of a collaboration between special-purpose or web-based simulators. For this type of simulators, usually the available computational power is a bottleneck to include additional complex mathematical calculations in real-time. Furthermore the simulators are only interested in the results of the physiological simulation. Therefore, a client-server result of the problem is very applicable. For this purpose one can establish a TCP socket server by using the QT classes <QTcpServer> and <QTcpSocket>.

5.1.4 Unit Testing

Unit testing is one of the important methods in modern programming, especially for big projects with many classes and in case of test-based software development. For this project unit testing is very important and crucial, because many classes were composed of others and hidden errors could therefore easily propagate from lower levels to higher ones. To avoid this problem, functional tests are mandatory. Furthermore, tests are necessary to guarantee the correct implementation of different components of the system, as given by figure 4.1, e.g. functional tests for XML parsing processes and quality tests for different solver functions. For this purposes two main libraries, <CppUnit> [162] and <QTest>[187], have been applied.

The test process itself is strongly related to definition of tests by XML descriptions. Because the models are described by XML as well. By this, functionalities of algebraic, semantic and differential solvers can be tested by standard software test routines. In the following, a simple XML model for a 1D ODE system and the corresponding functional and qaulity test is exemplary given, to demonstrate the strength of this notations and the impact for integrated testing.

```
% simple model for a 1D ODE system
<model>
  <name>ODE_1</name>
  <structure>
    <variables>
      <parm>
        <name>factor</name>
        <val>1</val>
      </parm>
      <var>
        <name>t</name>
        <val> 0.0 </val>
      </var>
      <var>
        <name>dt</name>
        <val> 0.01 </val>
      </var>
    </variables>
    <equations>
      <diffeq>
        <name>C</name>
        <integralname>C</integralname>
        <initialval> 1 </initialval>
        <dervvalue> -factor * C </dervvalue>
      </diffeq>
    </equations>
    <definitions>
      <block>
        <def>
          <name>t</name>
          <val> t+dt</val>
        </def>
      </block>
    </definitions>
  </structure>
    <display>
      <label>
        <labelvalue>C</labelvalue>
        <file>output.txt</file>
      </label>
    </display>
</model>
```

As one can see, the model is containing mainly one differential equation and additional algebraic definitions and equations. The result of the integration is saved to a log file ("output.txt")

by an export mechanism. Thus, the results can be used for functional unit testing afterwards, as demonstrated in the following, briefly.

```
class Test1dOde: public QObject
{
  Q_OBJECT
  private slots:
    void initTestCase(){
      physiosim.loadXML("1dOde.xml");
      physiosim.setODESolver("Rungekutta4");
    void Test(){
      physiosim.run(1000); //run 1000 steps
      SimResult = physiosim.import("output.txt",1); // import results
      RefResult = physiosim.import("ref.txt", 1); // import reference results
      for(int i=0; i<1000; i++){
        QVERIFY(SimResults[i] == RefResults[i]); }  // functional test
      }
    void cleanupTestCase(){
      physiome.cleanUp();
    }
    ...
 };
```

The numerical integration solver routines, applied in this project, are based on black box libraries. Therefore, additional quality tests can be applied to guarantee the maximal integration errors in comparison to analytical or reference integration methods. This is depicted for the code example of the ODE system before, in figure 5.1, where the Runge-Kutta-4 method and the implicit Euler method have been applied.

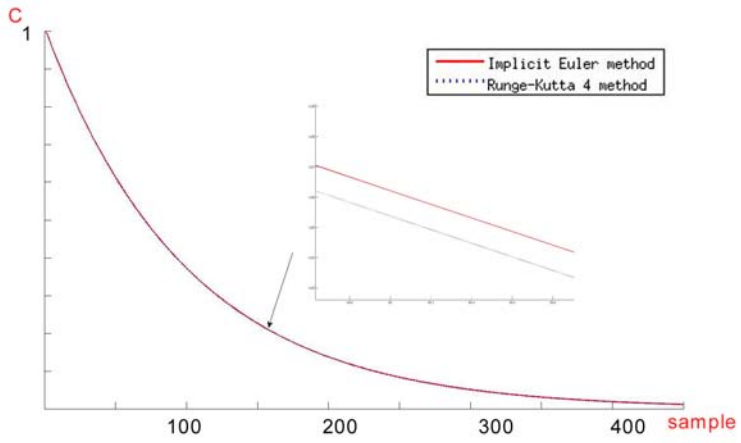


Figure 5.1: Results of a 1D ODE system, given by the XML code example before. One can see the small numerical integration error between implicit Euler method and the Ronge-Kutta4 method, which can play a significant role for more complex systems.

5.1.5 Maintenance

Due to the fact, that the proposed system and environment is first of all a proof of concept and the first version or prototype of the realized idea, it is very important to provide basic functionalities to enable easy maintenance for further-developments or integrations for future releases. For this purpose, a good documentation is mandatory, which is achieved by using the Doxygen [65] to generate human readable documentation of the source code.

Additionally, Subversion (SVN) [217] is used to provide functionalities for version control and cooperated development, allowing many developers to work on the same code by checkout a copy from the repository server and submit or commit their local changes to the global code on the repository .

This topic is also strongly related to optimization and multi-developer software development. Therefor, Hudson [105] - an extensible continuous integration software [67] - is applied to automate the build process. It supports the integration of SVN and plug-ins of automated unit testing and analysis.

The software development and unit-testing process itself is based on test-based design (referring to Blake Madden [162]) and is depicted in figure 5.2. The diagram shows the process flow of software component development. In the upper part, one can see the unit testing circuit, which clarify the hand in hand development of components and tests. If a component can be build and pass all tests, it enters the optimization circuit (lower part of the diagram), which includes the processes of version control, continuous integration and optimization, e.g. by profiling and quality assurance.

5.2 Hardware

The hardware components of this project are composed by a the MSP430 micro-controller series of ultra-low-power 16-bit RISC mixed-signal processors with 8 MHz clock frequency from Texas Instruments. This MSP430f16x microprocessor is usually used, which supports a wide range of tasks and methods, including among others digital I/O, timer, JTAG, serial communication modules as USART, digital analog conversion (DAC), analog digital conversion (ADC) and debugging [1].

[1] More information can be found in the data sheets on the Texas Instrument website : http://focus.ti.com/docs/prod/folders/print/msp430f169.html

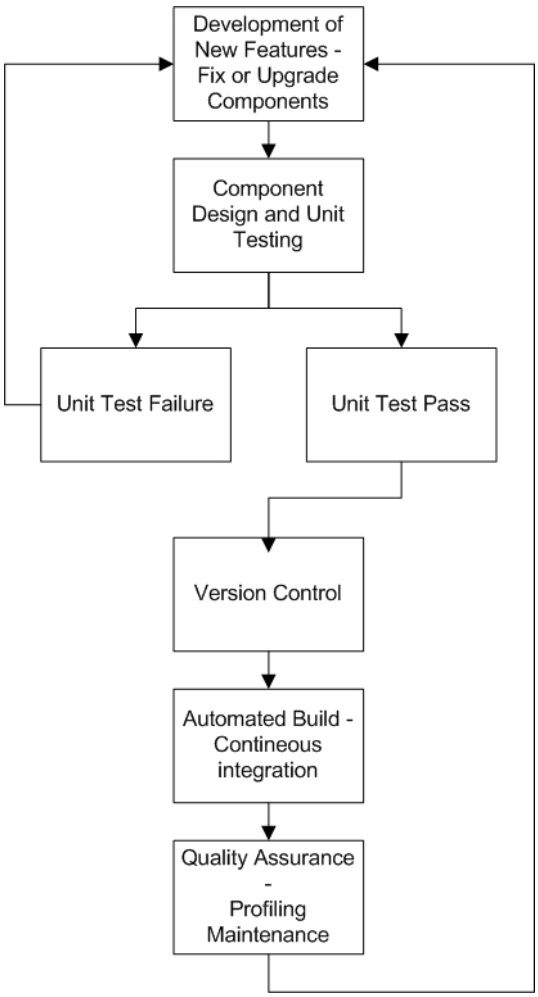


Figure 5.2: Test-based component development circuit: In the upper part one can see the unit testing circuit. Once a component passes all tests, it enters the optimization and maintenance circuit.

By combining the μC development environment with USB to UART conversion modules, one can design small and ultra low voltage embedded systems to act as clients for a diversity of client-server tasks, as will be shown in the chapter.

Due to the fact, that the μC is based on a RISC processor, one has to remind that parallel processing is not supported. Although C++ is used as programming language (supported by the IAR embedded workbench), during the compilation to machine code, it will take detour over serial C code. One has to kept this fact in mind to avoid deadlocks or starvation, especially when dealing with interrupt handling routines, which can abort main processing threads.

Therefore, a general semaphore protected processing loop is used, indicated by disabling and enabling the interrupt handling. The general processing flow is depicted in figure 5.3. As one

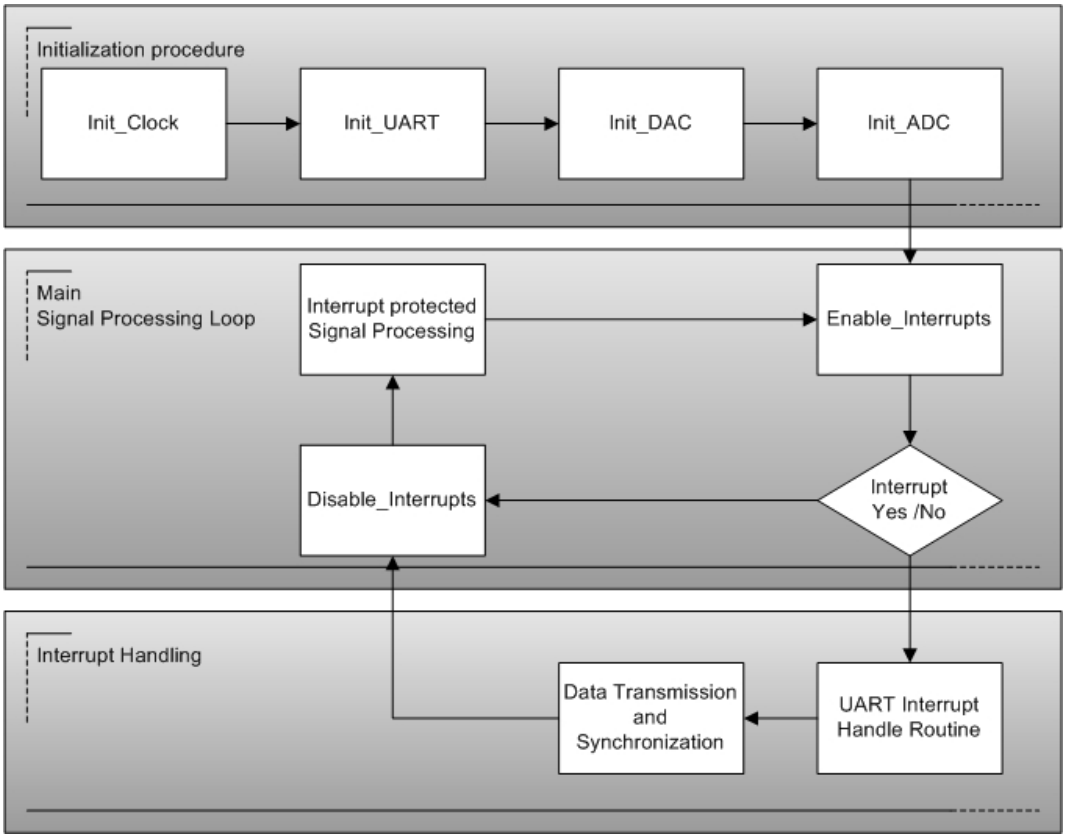


Figure 5.3: General processes flow on the client μC: During initialization the MCU and periphery of the μC are loaded. After this, one distinguish between the main process and the interrupt handling loops.

can see, first of all the MCU, the clock systems, I/O banks, UART, ADC and DAC registers are initialized. After that, the MCU inters the main processing and interrupt handling loops. Both are protected by hardware semaphore structures to guarantee a correct calculation.

The interrupt handling loop is mainly applied for I/O processing tasks, e.g. conversion and communication. The main processing loop contains the algorithmic realization of the clients, e.g. solving the forward problem of the ECG or the backward problem of the SaO2 emulation, as will be explained in chapter 6.

5.3 Resources

The software is realized by including various open-source and free C++ libraries as listed in table 5.2. Most of them are allowed to call themselves platform independent, however some nearly fit this attribute by supporting different compliers on different computer architectures.

Generally, the GNU C Compiler (GCC) Version 4.3 has been used for compilation and therefore it was possible to compile the source exemplary for two different platforms. A Gentoo Linux 64-bit and a Windows XP 32-bit operating systems. For the latter a minimalistic GNU for Windows (MinGW) was applied.

Name	Version	Description
STL	RT1	Standard template library, providing general constructs
PNL	1.0	Probabilistic Network Library
SMILE	1.1	Structural Modeling, Inference, and Learning Engine
GSL	1.13	GNU Scientific Library, for mathematic integration of ODE systems [87]
BOOST	1.39.0	Collection of free peer-reviewed portable C++ source libraries
QT	4.4	QT toolkit library for GUI application
Qwt	5.2	Qt extension including widgets for technical applications
log2cpp	0.3.4	C++ classes for flexible logging to files
CppUnit	1.12.1	Unit testing framework for C++
MinGW	5.1.6	Minimalist GNU for Windows

Table 5.2: Library resources

In addition to C++ libraries, many other useful tools have been used for the development of the system, as listed in table 5.3. For software development eclipse and QT-Creator have served as Integrated Development Environment (IDE). For the microprocessing programming the IAR embedded workbench IDE was used. The GUI have been designed based on QT using the QT-Designer, also due to binding of user defined interfaces into the simulation by definition of *.ui (QT's GUI file format in XML) files, the QT-Designer is important for further developments of new models.

For the design and development of BN models essentially the Genie software was applied, which provides a XML based representation of probabilistic networks, which can be loaded to the system, using the smile library.

Name	Version	Description
GCC	4.3	GNU C Compiler
Eclipse CDT	6.0.1	C/C++ integrated development environment
SVN[217]	1.6.6	Subversion, version control software
Genie	2.2	GUI for Bayesian network design
QT-Creator	1.0.0	IDE based on QT 4.5
QT-Designer	4.6.2	GUI designer based on QT
Hudson [105]	1.3.12	Continuous integration software
Doxygen [65]	1.6.2	Documentation system for programming
IAR IDE	5.2.9	Embedded workbench IDE for MSP430 C/C++ programming
Eagle	5.6	Cadsoft Easily Applicable Graphical Layout Editor

Table 5.3: Tool and IDE resources applied to realize the project.

6 Vital Signal Modeling And Emulation

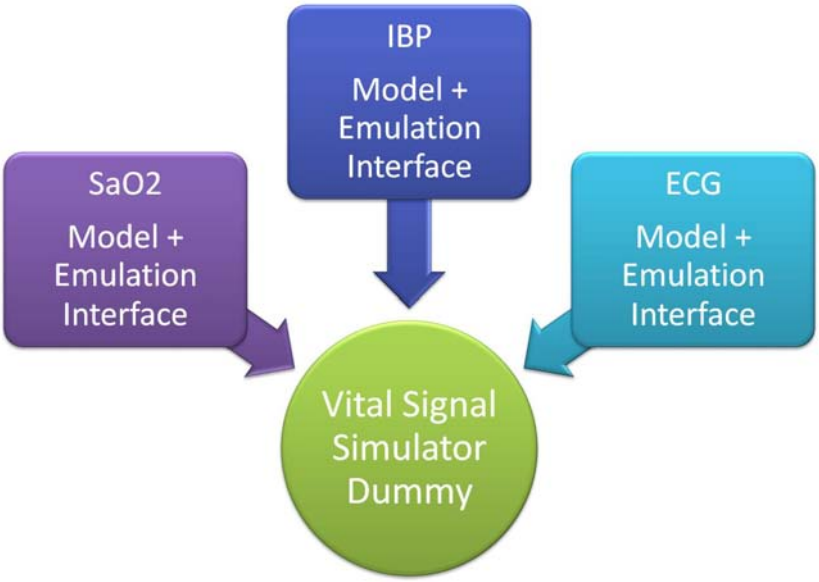


Figure 6.1: Vital signal emulation is based on novel modeling and inverse analysis of the corresponding vital signal sensing process. Thus, corresponding inverse models as well as the related hardware aspects are presented. Three emulation interfaces for ECG, SaO2 and IBP emulation (inverse sensing) are delivered, which can be integrated into a simulator dummy.

According to the definitions in "Functional anatomy of full-scale simulators" [232], there are various ways to describe interfaces between the simulation environments and the outer world. The syntax in literature, however, is unfortunately very inconsistent regarding the definition of such interfaces. For instance "direct connection of the internal electrical cables of the physiologic monitor" is considered as emulation, although the trainee is not using any real sensor and thus cannot understand the real process of physiological measurements.

During this work, we talk about emulation, if and only if the signals are emulated in a way, that arbitrary standard clinical monitoring equipment with standard sensor connections can be connected for measurement. Such interfaces consist of electrical or electro-mechanical actuators, as defined in chapter "System Architecture". For all the other forms of signal access (e.g. alpha-numerical, screen-based), we use the syntax of a simulation interface instead.

It is obvious that medical simulator systems will not move totally towards full VR environments - at least as long as we have no virtual haptics - because the trainees need to feel a real (or as real as possible) clinical situation. Additionally, for physiological full-scale simulators or those part-scale simulators, where partially physiological signals are necessary, it makes a big difference to have a real emulation of the signals, such that arbitrary monitoring equipment can be attached, or to display the parameter on a display (screen-based simulation) [49][62].

One argument for this difference is, that if the personal is able to use the same clinical environment as in real praxis - and this includes the use of the same diagnostic monitoring equipment - the artificial training situation is more close to a real case scenario. Advantages of vital signal emulation in contrary to screen based simulation (e.g. more effective educational effects) have already been presented in chapter "State Of The Art" in detail.

ECG and pulseoximetry (SaO2) are the essential tools in non-invasive medical sensoring for diagnostics [201]. Also the (minimal) invasive blood pressure (Invasive Blood Pressure (IBP)) measurements are standard in critical health care data monitoring. Nevertheless, in practice **these parameters were not emulated in an adequate way until now**.

Therefore, in the next sections, substantial methods and models necessary for emulation of the above signals (ECG, SaO2, IBP) will be presented, which can be integrated into a target simulation dummy, as depicted in figure 6.1. In order to be able to deliver the necessary hardware devices, one has to analyze each medical vital signal sensing process independently and has to find inverse solutions.

Thus, the mathematics and the models behind these problems are essentially targeting **inverse problems** and the author is glad to have contributed to the domain of vital signal modeling as well. Especially, the proposed ECG model can be used not only for ECG simulation or emulation, but also for quantitative electro-physiological description [5].

6.1 Validation

For the validation of the functionality of the emulators, a closed loop system has been used. For this purpose the emulated signals where detected by a patient monitor instrument, and the data were collected by a central monitoring software, especially designed for this purpose. Then the difference between nominal and actual values could be calculated as a measure for the quality of the emulation. This system is demonstrated in figure 6.2.

The patient monitor applied for the validation was a Tianrong TOMORROW TR-910 [225]

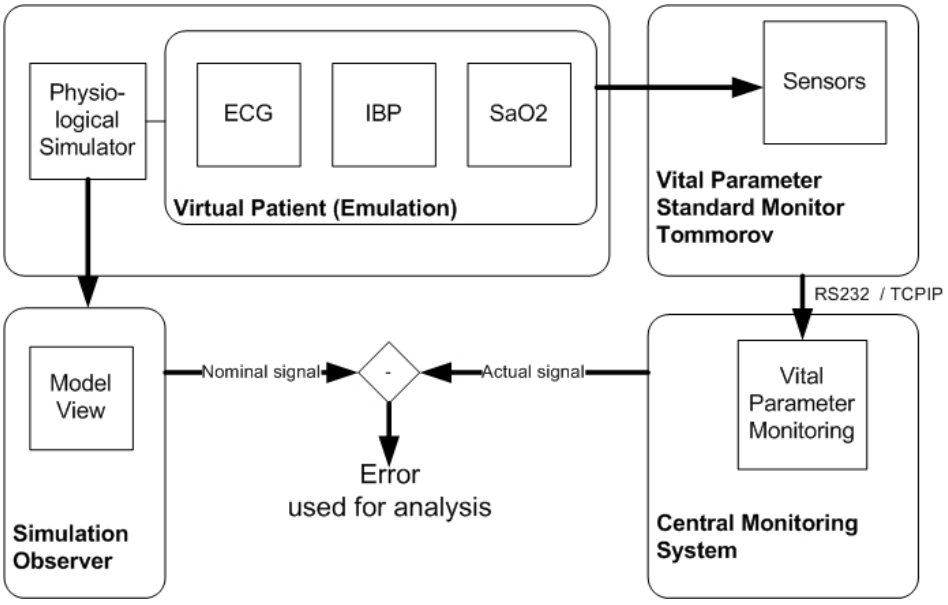


Figure 6.2: Validation of emulation hardware interfaces

monitor. The data were transported by serial RS232 or Ethernet cable according to a serial transmission protocol. Additionally Phillips patient monitor systems have been used to proof the general functionality of the proposed environment by a second system.

The system proposed in figure 6.2 could actually be used very effectively for reliability analysis of patient monitoring systems. Due to the closed relation of simulation and reliability or dependability analysis, this topic is discussed separately at chapter 9 at page 159.

6.2 IBP Emulation Interface

According to the knowledge of the author, the emulation of invasive blood pressure for measurements is not supported by full or part scale patient simulators yet. Therefore, in the following a system to support pulsative pressure wave generation with defined shape, amplitudes and frequencies, will be presented.

6.2.1 System Analysis

First of all, a system analysis procedure has been applied for the hydraulic pipe and a pressure generator, as depicted in figure 6.3. A single-acting hydraulic cylinder is used as a pressure

generator, driven by a linear motor driver. The hydraulic pipe or connection is defined by a length l_k and a cross-section A_k.

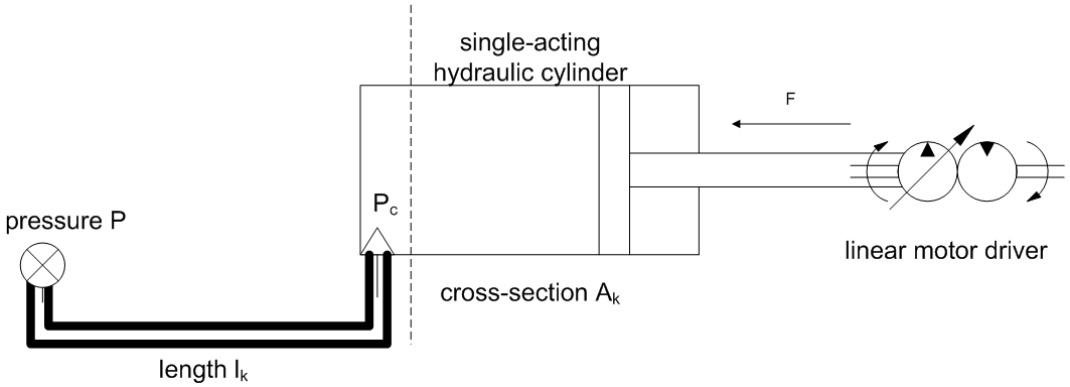


Figure 6.3: IBP prototype system for analysis. A pipe is used representing the invasive catheter access to the vascular system. This pipe is characterized by a length l_k and a cross section A_k. Usually, the catheters are flushed with sodium chloride solution before measurement, to prevent the appearance of elastic air bubbles inside the pipe. Therefore, the pipe is assumed to be filled with water. A pressure wave P_c at the valve of the cylinder is then transported by the pipe, resulting in a pressure P at the sensor. The pressure wave P_c is generated using a linear motor driver.

For the pipe system an equivalent electrical circuit can be used to describe the dynamics, as shown in figure 6.4.

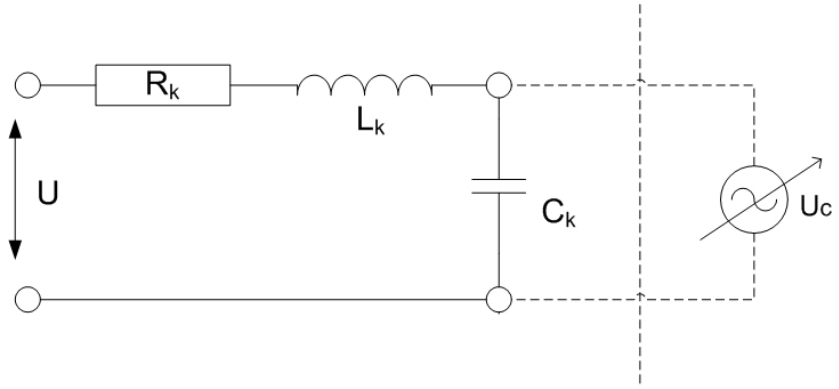


Figure 6.4: IBP pipe system equivalent electrical circuit. A voltage generator is generating the input of the system U_c. The output U is depending on resistance R_k, inductance L_k and capacitance C_k.

R_k, L_k and C_k are resistance, inductance and capacitance of the pipe. The resulting input-

output relation is thus given by equation 6.1:

$$\begin{aligned} \frac{U_c}{U} &= \frac{\frac{1}{iwC_k}}{\frac{1}{iwC_k} + iwL_k + R_k} \\ &= \frac{1}{1 + iwCkRk - w^2 C_K L_k} \end{aligned} \tag{6.1}$$

The system transfer function $G(iw)$ is thus given by:

$$G(iw) = \frac{K}{1 + 2dTiw + T^2(iw)^2} \tag{6.2}$$

for $K = 1$, damping $d = \frac{R_k C_k}{2T}$ and $T = \pm\sqrt{L_k C_k}$.

In this case, we are interested in the limiting frequency f_0 of the system, which should be beyond the main frequency of the pressure driver. For $w_0 = 2\pi f_0 = \frac{1}{T}$, this is given by:

$$f_0 = \frac{1}{2\pi\sqrt{L_k C_k}} \tag{6.3}$$

The hydro-mechanical equivalent parameter for electrical resistance is the flow resistance R_f, which is defined according to Hagen-Poiseuille law as following :

$$R_f = \frac{8\eta l_k}{\pi r_k^4} \tag{6.4}$$

while η is the dynamic fluid viscosity, l_k is the pipe length and r_k the radius of the pipe, assuming homogeneity for fluid and pipe. The effective mass m_f is representing the inertia (fluid mechanic analog of L_k), which is defined by $m_f = \frac{\rho l_k}{A_k} = \frac{\rho l_k}{\pi r_k^2}$, for the fluid density ρ and the cross-section of the pipe A_k. The pipe compliance C_f is the analog for the electrical capacitance C_k.

This leads to the following damping d and cut-off frequency f_0:

$$f_0 = \frac{1}{2\pi\sqrt{\frac{\rho l_k}{\pi r_k^2} C_f}} = \frac{r_k}{2\sqrt{\pi\rho l_k C_f}} \tag{6.5}$$

$$d = \frac{R_f}{2}\sqrt{\frac{C_f}{m_f}} = \frac{4\eta}{r_k^3}\sqrt{\frac{l_k C_f}{\rho\pi}} \tag{6.6}$$

The analysis result for the cut-off frequency is demonstrated in figure 6.5.

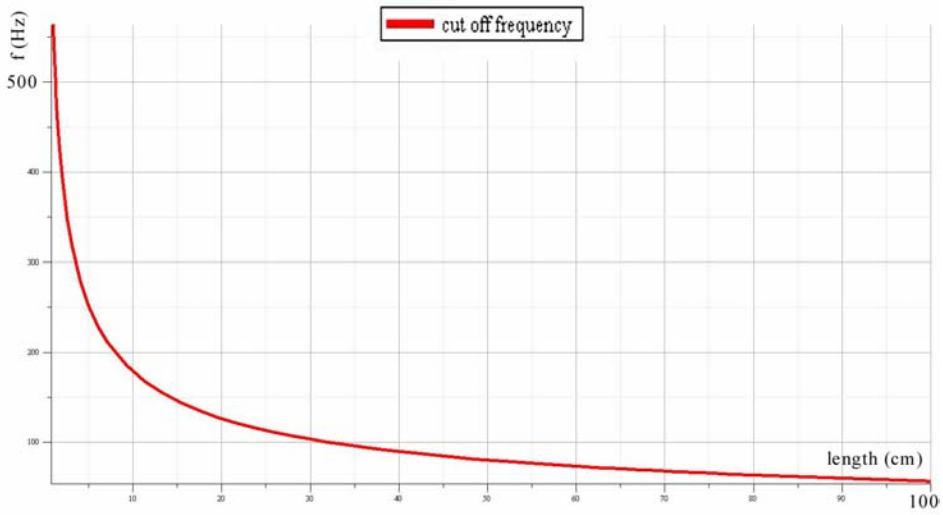


Figure 6.5: Cut off frequency f_0 (in [Hz]) in dependency to length l_k (in [cm]). Following assumptions were applied: $\rho = 10^{-3} kg/cm^3$, $r = 0.02cm$, $\eta = 10^{-3} Pa.s$, $C_f = 10^{-7} cm^3/Pa$.

As can be seen, the cut off frequency for the pipe length used (60-100 cm) in praxis is faraway from the critical region. The typical frequencies for the IBP waveform signals are in the region of 0-10 Hz.

6.2.2 Prototyping

The system composed in figure 6.3 was realized after positive simulation and system analysis prototypically, as depicted in figure 6.6

Although, the system could satisfy the emulation purpose, we noticed several problems with the current design:

- Some components, e.g. the mechanical coupling, were not off-the-shelf components.

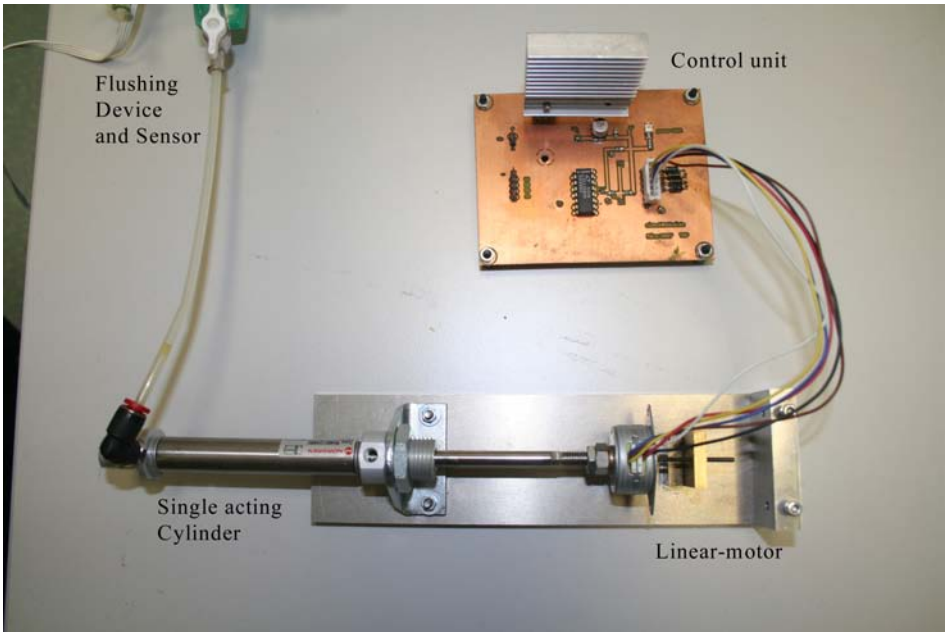


Figure 6.6: IBP prototype realization. One can see the cylinder (Norgren RT/57120/M/50) and linear motor (LSP2575M0506-M2X0,4) on the bottom. The pipe leaves the single-acting hydraulic cylinder at the left and is connected to the flushing device and IBP sensor. On the top right a linear motor control hardware (based on MSP430, LTC1156 and TL750L) is depicted.

- The hydraulic cylinders are made for special oil fluids. Contact with water could change some attributes, e.g. variation of friction of the cylinder piston over time was observed.

- For the prototype, a self made thread was used to transform the rotational motion, created by the linear motor, into a translating one. This mechanical coupling is very sensitive to abrasion. To reduce friction an oil basin should be used.

- Due to mechanical movement of the piston and the linear motor axis, the setup was heated.

- The suggested system was realized in open loop mode.

- Due to mechanical motion the setup was noisy during runtime.

To overcome the above problems, the components would become very expensive. Therefore, alternative solutions were analyzed with the idea in mind, to be able to provide off-the-shelf solutions. To avoid mechanical coupling, one can use a membrane pump, but this solution is neither inexpensive nor off-the-shelf. Therefore, an alternative method based on pneumatic control was chosen, which will be presented next. Assuming the pipe is filled with water, the

previous system analysis is still valid for the new compositon.

6.2.3 System Design

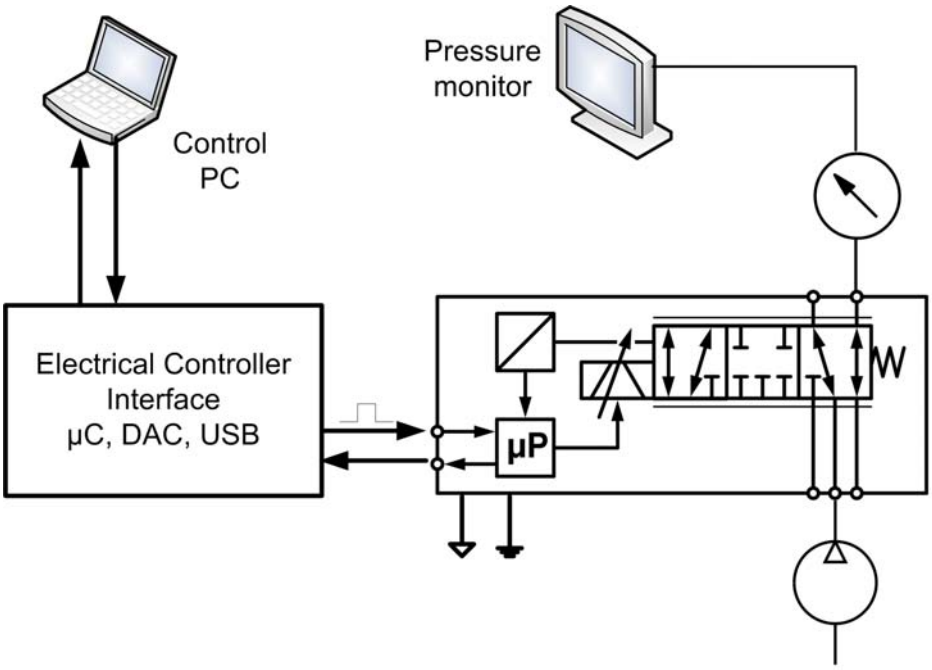


Figure 6.7: IBP emulator system overview. This system is based on pneumatic pressure control in closed loop mode.

For the purpose of systemic pressure generation an air pump has been used (Aqua Medic Mistral 300[1]), which is capable to generate a maximal air pressure of 470 mbar. Of course the system is not restricted to this pressure generator and could be also supplied by other air pumps or air compressors with similar maximal pressure and a fan efficiency of approximately 300 l/h.

For control of pulsatile waves a 5/3 way proportional valve (Norgren VP60[2]) has been used. By this a linearized bidirectional flow control valve can be realized. For low flows this allows a fast and precise closed loop control of small pressures (<=1 bar). The system design is schematically depicted in figure 6.7.

[1] http://www.aqua-medic.de/seawater/de/14/Mistral/
[2] http://www.norgren.com/

6.2.4 Characteristics And Results

In figure 6.8, one can see the system's response to step functions. Additionally, in figure 6.9 a sinus function with period $T = 1s$ was used to drive the system.

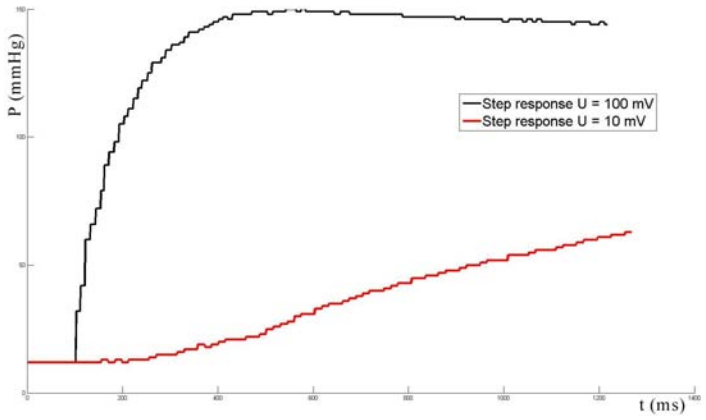


Figure 6.8: System step response. The pressure P is voltage controlled. One can see the step response for two different control voltages $U = 100mV$ and $U = 10mV$

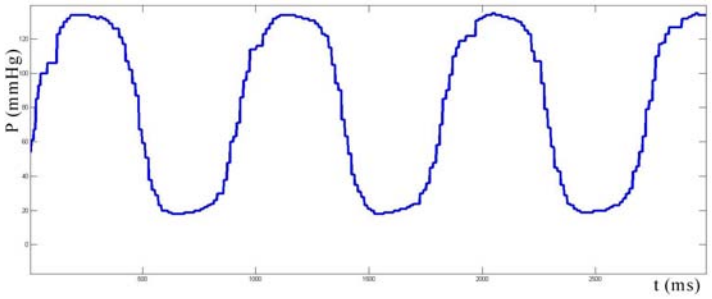


Figure 6.9: Sinus response. The pressure P is voltage controlled. One can see the response to a sinus function with period $T = 1s$.

The realization of the suggested system shows a very realistic way to emulate blood pressure, invasive measured over a catheter entrance. The signals can then be detected by regular blood pressure measurement systems and standard patient monitoring equipment. As to the knowledge of the author, this is unique among existing simulation interfaces.

In figure 6.10 one can see the signals, measured by a sensor, which was attached to the IBP emulation interface.

The system and its components are depicted in figure in detail 6.11. The novelty of the setup is, that one is able to attach a blood pressure transducer directly to the catheter, which is connected to the output pipe from the 5/3-way proportional valve controller (VP60). The

emulated pressure waves are very realistic and not to distinguish from real patient pressure waves.

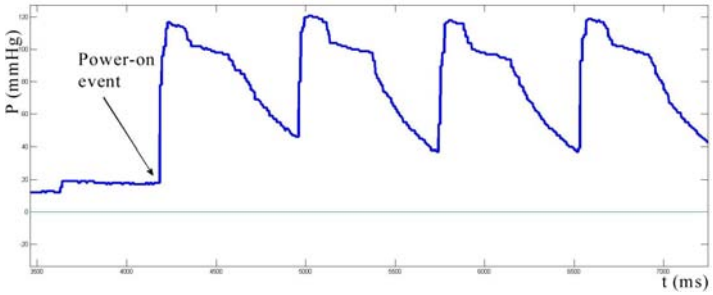
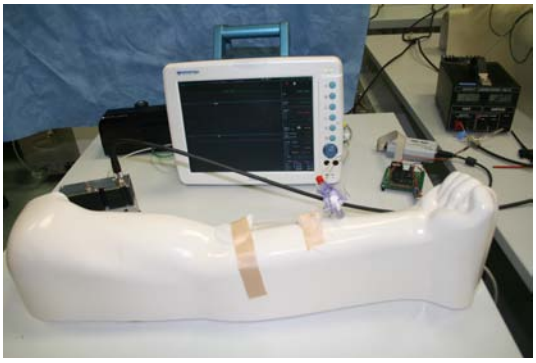
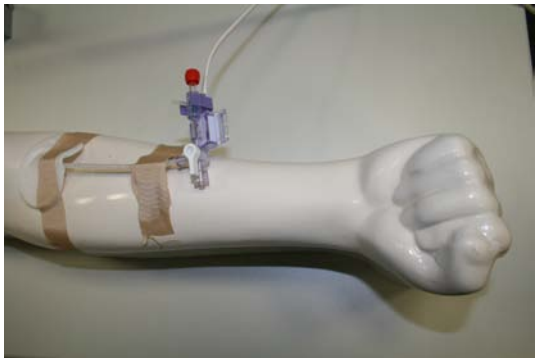


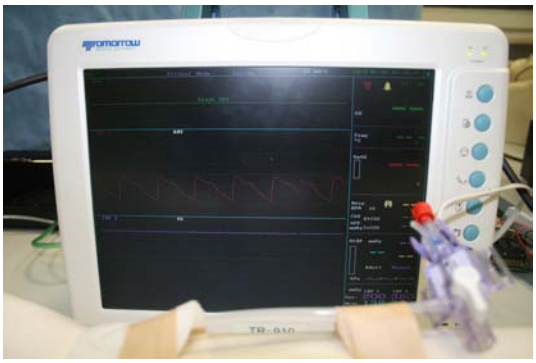
Figure 6.10: IBP emulated signal. Measured with TOMORROW patient monitor system. The arrow is marking the power-on event. Before this time, the device is not controlled and is therefore in safe mode. The control valves are open, to prevent any damage to the sensing and emulation equipments.



(a) All IBP emulator system components



(b) Catheter for attachment of the blood pressure transducer



(c) Signal measured on standard patient monitor

(d) 5/3 way proportional valve controller

(e) MSP430 micro controller for regulation

(f) Mistral 300 and Norgren VP60

Figure 6.11: IBP emulation hardware interface system.

6.3 SaO2 Emulation Interface

6.3.1 SaO2 Principles

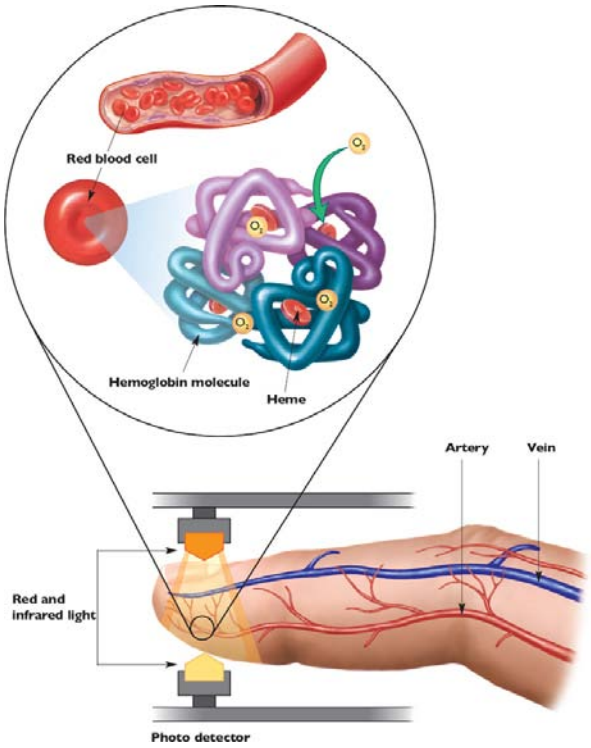


Figure 6.12: Principle of pulseoximetry: Different substances have different wavelength specific extinction coefficients. This principle formulated by the Lambert-Beer law can be used to measure the pulsation of blood and the saturation of oxygenated hemoglobin in blood.

To emulate the necessary signals for a SaO2 sensor, one has to know and understand the principle [150][241] of pulsoximetry. Then, an inverse system analysis can lead to sophisticated emulation of the signal, which can be measured by a standard SaO2 sensor and monitor. Principally, emulation of pulsoximetry means to simulate tissue reaction to incident light.

We focus on light based pulsoximetry, which is commonly established as an noninvasive patient vital signal sensing procedure in medical everyday procedures since the 1990s. The strategy is depicted in figure 6.12.

$$SaO_2 = \frac{c(Hb)}{c_{total}} = \frac{c(Hb)}{c(Hb) + c(HbO_2)} \tag{6.7}$$

6.3. SAO2 EMULATION INTERFACE

The SaO_2 is describing the percentage of hemoglobin molecules in blood, which are carrying their full potential of oxygen [228]. Oxygenated hemoglobin (HbO_2) absorbs more infrared light and desoxygenated hemoglobin (Hb) absorbs more red light. Therefore, as shown in figure 6.12, red and infrared light are transmitted by a Light Emitting Diode (LED) on the one side of the tissue (e.g. finger or earlobe) and the intensity of light after absorption by the tissue is received with photo detectors, on the other side.

The relationship between measured light absorption and saturation is related to the percentage of oxygenated hemoglobin to the total amount of hemoglobin. Although empirically calibrated and only measuring with two wavelengths, pulsoximeters have shown accuracy and usefulness in clinical evaluation and have become standard for routine monitoring [21]. Studies in human volunteers proof the good performance of pulsoximetry devices for healthy adults in a saturations range (SaO_2) of 70 to 100 %.

The intensity, which is measured after absorption, can be described according to the well-known Beer-Lambert law (equation 6.8).

$$I_{out} = I_{in} e^{\sum_{i=1}^{N} \epsilon_i c_i d_i} \tag{6.8}$$

While I_{in} is the intensity of light before entering and I_{out} after leaving the tissue, ϵ_i is the extinction coefficient, and c_i is the concentration for $i = 1..N$ different absorbing materials and the corresponding thickness d_i. Usually, the only time-varying thickness is given by the pulsation of arteries, which have an elastic wall and can thus change in diameter (Windkessel effect).

In the case of functional pulsoximetry, only two different light absorbing substances are interesting, Hb and HbO_2. The effect of other substances, e.g. tissue or bone, has to be filtered out by computation. First measurements are taken at two different times. This sampling is applied within a sampling rate of 1 KHz. By this method the effect of constant not time-varying absorption will be filtered out. For this purpose, the signal composition is assumed as described by equation 6.9.

$$I_{out} = I_{in} e^{\epsilon_\mu c_\mu d_\mu + \alpha(\lambda) d(t)} \tag{6.9}$$

while ϵ_μ, c_μ and d_μ describe the extinction coefficient, concentration and thickness of non-changing substances like tissue and bone, and $\alpha(\lambda) = \sum_{i=1}^{N} \epsilon_i(\lambda)c_i$ describe the absorption coefficient for N non-constant components and $d(t)$ the time varying thickness. λ describes the wavelength of the incident light, resulting in a specific extinction $\epsilon_i(\lambda)$ and thus in a specific absorption $\alpha(\lambda)$.

Then, measurements at two different times t_1 and t_2 will lead to:

$$\begin{aligned} \frac{I(\lambda, t_1)}{I(\lambda, t_2)} &= \frac{I_{in} e^{\epsilon_\mu c_\mu d_\mu + \alpha(\lambda) d(t_1)}}{I_{in} e^{\epsilon_\mu c_\mu d_\mu + \alpha(\lambda) d(t_2)}} \\ &= e^{\alpha(\lambda)(d(t_1) - d(t_2))} \end{aligned} \tag{6.10}$$

According to equation 6.10, constant terms like incident light intensity and constant absorptions are canceled, by this method. However, the absorption is still related to the time-varying thickness. This is not dependent on the wavelengths of the incident light, in contrary to the absorption coefficient and thus can be canceled by measurements at two different wavelengths λ_1 and λ_2, resulting in the following formula:

$$\begin{aligned} \frac{\frac{I(\lambda_1, t_1)}{I(\lambda_1, t_2)}}{\frac{I(\lambda_2, t_1)}{I(\lambda_2, t_2)}} &= \frac{e^{\alpha(\lambda_1)(d(t_1) - d(t_2))}}{e^{\alpha(\lambda_2)(d(t_1) - d(t_2))}} \\ &= \frac{e^{\alpha(\lambda_1)}}{e^{\alpha(\lambda_2)}} \end{aligned} \tag{6.11}$$

By applying the natural logarithm, we have a measured parameter Ω (compare equations 6.12), based on measurements with two different wavelengths λ_1 and λ_2 at two different times t_1 and t_2, which is perfectly describing the relation to the absorption-coefficient of non-constant components like Hb and HbO_2 in arterial blood. Those are essentially dependent on the concentrations c_{Hb} and c_{HbO_2}, which we are interested in.

$$\begin{aligned} \Omega &= \frac{ln \frac{I(\lambda_1, t_1)}{I(\lambda_1, t_2)}}{ln \frac{I(\lambda_2, t_1)}{I(\lambda_2, t_2)}} \\ &= \frac{\alpha(\lambda_1)}{\alpha(\lambda_2)} \\ &= \frac{\epsilon_{Hb}(\lambda_1) \cdot c(Hb) + \epsilon_{HbO_2}(\lambda_1) \cdot c(HbO_2)}{\epsilon_{Hb}(\lambda_2) \cdot c(Hb) + \epsilon_{HbO_2}(\lambda_2) \cdot c(HbO_2)} \end{aligned} \tag{6.12}$$

The extinction coefficients for different wavelengths are known and thus the measurement parameter Ω is directly related to the concentrations $c(Hb)$ and $c(HbO_2)$. The formal relation to the searched value $SaO2$ (according to 6.7) can be established by the equitations 6.13. For each absorption coefficient we can reformulate the relation for $c_{total} = c(Hb) + c(HbO2)$, using mathematical expansions in "[]":

$$\begin{aligned}\alpha &= \epsilon_{Hb} \cdot c(Hb) + [\epsilon_{Hb} \cdot c(HbO_2)] + \epsilon_{HbO_2} \cdot c(HbO_2) - [\epsilon_{Hb} \cdot c(HbO_2)] \\ &= \epsilon_{Hb} \cdot (c(Hb) + c(HbO_2)) + (\epsilon_{HbO_2} - \epsilon_{Hb}) \cdot c(Hb0_2) \left[\frac{c(Hb) + c(Hb0_2)}{c(Hb) + c(Hb0_2)}\right] \\ &= \epsilon_{Hb} \cdot c_{total} + (\epsilon_{HbO_2} - \epsilon_{Hb}) \cdot SaO_2 \cdot c_{total} \\ &= c_{total} \cdot (\epsilon_{Hb} + (\epsilon_{HbO_2} - \epsilon_{Hb}) \cdot SaO_2) \end{aligned} \tag{6.13}$$

Thus, equation 6.12 results in:

$$\Omega = \frac{\alpha(\lambda_1)}{\alpha(\lambda_2)} = \frac{\epsilon_{Hb}(\lambda_1) + (\epsilon_{HbO_2}(\lambda_1) - \epsilon_{Hb}(\lambda_1)) \cdot SaO_2}{\epsilon_{Hb}(\lambda_2) + (\epsilon_{HbO_2}(\lambda_2) - \epsilon_{Hb}(\lambda_2)) \cdot SaO_2} \tag{6.14}$$

and reformulation leads to:

$$SaO_2 = \frac{\epsilon_{Hb}(\lambda_1) - \Omega \cdot \epsilon_{HbO_2}(\lambda_2)}{\Omega \cdot (\epsilon_{HbO_2}(\lambda_2) - \epsilon_{Hb}(\lambda_2)) - (\epsilon_{HbO_2}(\lambda_1) - \epsilon_{Hb}(\lambda_1))} \tag{6.15}$$

6.3.2 Inverse Pulsoximetry

As can be seen from equation 6.15, the functional oxygen saturation SaO_2 is a function of measurements Ω and constant known extinction coefficients. Based on this theorem, most of the pulsoximeter devices are realized [39] in praxis, using two different wavelengths $\lambda_1 = 660nm$ and $\lambda_2 = 940nm$. These wavelengths are chosen, because the characteristic extinction coefficients - at these two wavelengths - for oxygenated and desoxygenated hemoglobin lead to maximal signal to noise ratio (SNR).

Although the relation between $SaO2$ and Ω is exactly described by equation 6.15, usually the relation is reduced to a hyperbolic relation, given by calibration values [39]. For an oxygen

saturation above 80%, which contains the relevant clinical information, the relation can even be approximated as linear. In summary, the dependency has the form $SaO_2 = f(\Omega)$ and in the simplest case $SaO_2 = a \cdot \Omega$.

The function f is statically defined by the vital monitoring systems. Therefore, the variation of the pulsoximeter signal, which is the purpose of an inverse pulsoximeter or a pulsoximeter emulator, has to be achieved by modifications of the parameter Ω, which is determined by equation 6.12.

To reduce the influence of noise and artifacts during the monitoring of pulsoximetry, modern systems do not take only two different time measures to calculate the information. This would lead to a very noise sensitive device. Therefore, to cancel the absorption effect of time-constant parameters (DC components), in praxis band pass filters are used. The non-static and time varying signal (AC component) is also determined by this, leading to the following formula:

$$\Omega = \frac{ln(AC(\lambda_r))}{ln(AC(\lambda_{ir}))} \tag{6.16}$$

while $AC(\lambda_r)$ and $AC(\lambda_{ir})$ are the dynamic ranges of the AC components (time varying signals) for the red and infrared light measurements. We assume a periodic composition of the emulated signals for red and infrared components, as given by equation 6.17, for heart rate HR, constant components $const_r$ and $const_{ir}$ and amplitudes A_r and A_{ir} at different wavelengths for red (r) and infrared (ir).

$$\begin{aligned} I(r) &= const_r - A_r sin(2 \cdot \pi \cdot HR \cdot t) \\ I(ir) &= const_{ir} - A_{ir} sin(2 \cdot \pi \cdot HR \cdot t) \end{aligned} \tag{6.17}$$

Then, according to equation 6.16 Ω is given by:

$$\Omega = \frac{ln(AC(\lambda_r))}{ln(AC(\lambda_{ir}))}$$
$$= \frac{ln(A_r sin(2 \cdot \pi \cdot HR \cdot t))}{ln(A_{ir} sin(2 \cdot \pi \cdot HR \cdot t))} \tag{6.18}$$
$$= \frac{ln(A_r)}{ln(A_{ir})} \tag{6.19}$$

In our approach, we keep the amplitude for the red signal A_r constant and only modify the infrared signal amplitude A_{ir}. One has to consider, that the signal shape itself, which was assumed as a sine function in equation 6.17 for simplification, can have a more complex time varying progress. This signal is described by the pulsation of the arterial vessels, which results in the time varying pulsation signal of pulsoximeters. Usually, the red AC signal component is used to extract the blood pulsation, which can be applied to calculate the HR on pulsoximetry systems.

Therefor, the signal shape for the red signal was approximated by the following functions and conditions:

$$I(r, \phi) = const_r - \sum_i^N A_i e^{\frac{-(\phi - \mu_i)^2}{2 * \sigma_i^2}}$$
$$\phi = \frac{2 \cdot \pi}{T} \cdot t \in [0..2\pi]$$
$$I(r, 0) = I(r, 2\pi)$$
$$t \in [0..T] \tag{6.20}$$

Usually $N = 2$ Gaussian functions (described by amplitude A_i, mean μ_i and standard deviation σ_i) are sophisticated to approximate the arterial pulsation curve, determined by systole and diastole phases. T is describing the period of the pulsation which is related according $T = 1/HR$ to the heart rate (HR). ϕ is thus defining the way on a circular orbit, where the Gaussian functions are defined and the mean values μ_i define the peaks of the systole and diastole components of the blood pulsation curve.

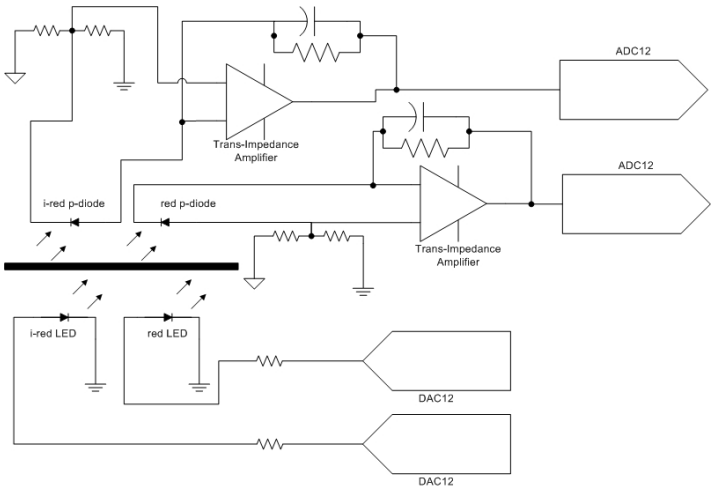


Figure 6.13: Emulation system for pulsoximetry (inverse pulsoximeter): The system is composed by two layers. The top layer is the sensor layer, sensing the incident light at two different wavelengths (including amplification and sampling). The bottom layer is the actuator layer, which emits light for two different wavelengths, controlled by a μC according to the theory of inverse pulsoximetry (equations 6.18 and 6.15) depending on the informations from the sensor layer.

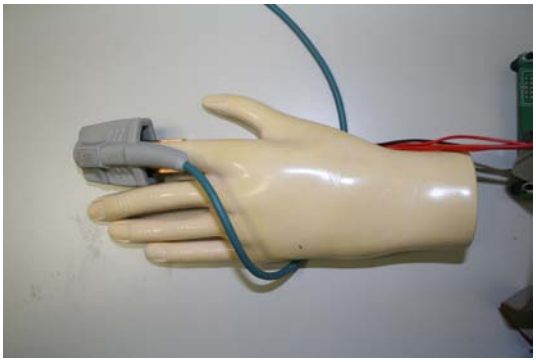
6.3.3 Characteristics And Results

According to the theoretical consideration of inverse pulsoximetry emulation (equations 6.18 and 6.15), a hardware system, based on the principles of system architecture (see also chapter 4), is used to realize the required algorithms with sensors and actuators.

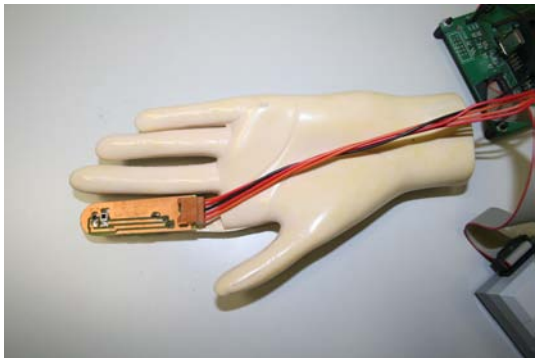
To mimic time varying absorption of blood, one needs therefore to vary the absorption for two different wavelengths, which could also be done by a mechanical device, e.g. by a rotating disc, with different absorbing areas [204]. Such systems are used to calibrate pulsoximeters in laboratories, but due to their mechanical nature, they are less optimal for simulation purpose (compare problems with mechanical simulation of blood pressure in section 6.2).

Therefore, in our approach, we emulate the necessary behavior without any mechanical parts, but only with electronic signal analysis and processing by the meaning of mimicking different intensities for different wavelengths and different times. A similar idea was proposed by Sweitzer et al. [219] in a patent formulation to simulate such signals for remotely located patients, which only describes the overall system for such an "oximetry adapter" and not any underlying theory or detailed realization.

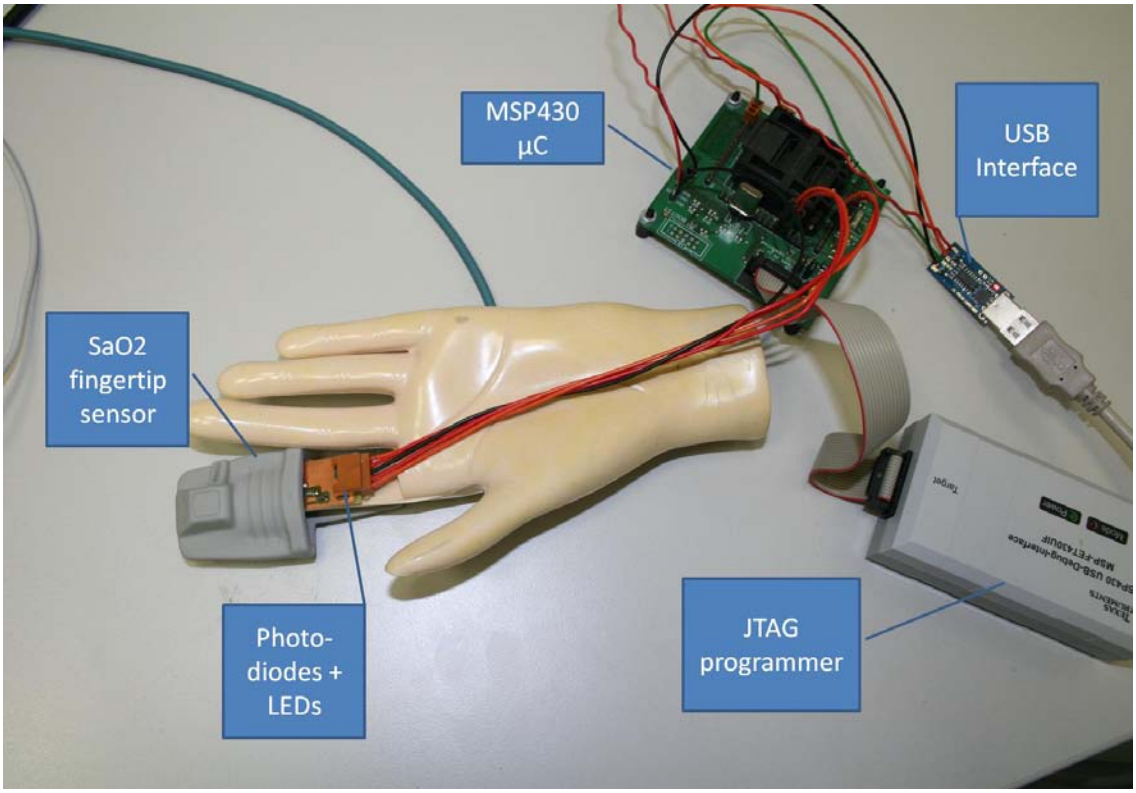
Many companies, especially focusing on medical device quality assurance, also offer products using the term "Oximetry Simulator". Unfortunately, these products, which mainly target



(a) The fingertip pulsoximetry sensor placed on the emulator for measurement.



(b) The emulator, representing a finger. It consists of actuators and sensors according to figure 6.13



(c) All pulsoximetry emulation system components.

Figure 6.14: SaO2 emulation hardware interface system. One can see the MSP430 μ-C development kit and the JTAG programmer for rapid prototyping, the communication interface with USB, the actor and sensor emulator finger, and the real sensor fingertip, which can be placed over the artificial inverse pulsoximetry finger.

medical sensor testing equipment, first do not support any information about the technical realization, and second provide limited functionalities, e.g. limited number of test cases by only four saturation rates and three pulse rates (HR) [185].

By the general theorems offered in this work, continuous emulation of SaO2 signals for arbitrary pulse and saturation rates is possible. The system composition to realize the inverse pulsoximeter is shown in figure 6.13.Two layers are representing the actuators and sensors. The wavelengths specific photo diodes are used in trans-impedance amplification mode. The signals are sampled with the MSP430 μ-C, which also controls the light emission of the actuator LEDs.

The electrical system schematics are provided in the appendix by figures 12.7, 12.6 and 12.5, for the interested reader.

The prototype of the system is depicted in figure 6.14. As one can see, the pulsoximetry finger tip can be attached to an artificial (inverse pulsoximetry) finger.

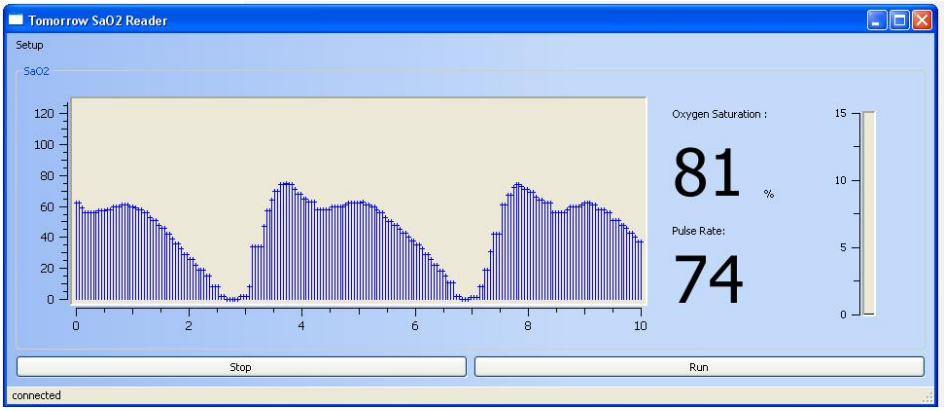


Figure 6.15: Simulated SaO2 signal, measured by a Tomorrow TR-910 patient monitor.

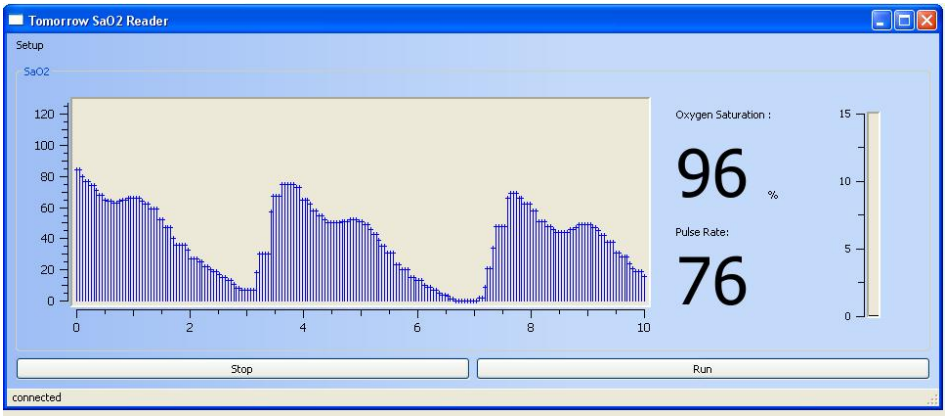


Figure 6.16: Real SaO2 signal, measured by a Tomorrow TR-910 patient monitor.

In comparison to industrial SaO_2 emulation devices like OxSim [185], where only static pulses and a minimal set of different oxygen saturation levels can be sensed, by our approach, we are capable to simulate all forms of clinical relevant signals. Thus, the resulting system is

much more realistic and will provide a new standard in medical simulator hardware for SaO2 emulation. Of course, by this system, we also can target medical sensor testing and reliability analysis.

In figure 6.16 an example of a real measured SaO2 signal is pictured. In figure 6.15 one can see the signal which is measured by emulation, according to the setup, depicted in figure 6.14. Real and emulated signal shapes are not to distinguish and thus very realistic. The measured signals are recorded by the remote monitoring tools, especially designed for this purpose.

6.4 ECG Emulation Interface

During the next subsection, a novel methodology for modeling ECG signals and waveforms and simulating the electrical potentials by software and hardware is presented. The approach is based on multiple dipoles, compared to standard single dipole approaches, known from vector electrocardiography.

The multiple dipole parameters are derived from real patient data (e.g. four dipoles from 12-channel ECG) by solving the backward problem of ECG numerically. Results are transformed to a waveform description, based on Gaussian mixture, for every dimension of each dipole. The Gaussian parameters show a sophisticated representation for known electro-physiological attributes e.g. ST-length or the QRS-complex and can therefore be used for characterization of anomalies.

These compact parameterized descriptors are used for a very realistic real-time simulation applying the forward solution of the proposed model with a specialized hardware.

6.4.1 System Design

The system design is given by a client server model. The server is a computer system capable to calculate the forward problem of the ECG for a multiple dipole model. The communication with the client is based on USB and the client is also powered over USB itself. The client is realized by a two layered PCB board, which was designed for this purpose specifically, based on three main components, USB communication interface, 16 bit RISC μC and DAC part for each electrode.

The system composition (schematic) is depicted in figure 12.4 in the appendix for the interested reader.

6.4.2 Model Characteristics

The multiple dipole model applied for ECG simulation is based on one or more electrical sources. The sources are modeled as electrical dipoles $d_m(t)$ with an origin, a length and a direction. Then the electrical potential field - the so called far-field - can be approximated by equation 6.21, which is the forward solution of the overall electro physiological model.

$$\phi(r,t) = \frac{1}{4\pi\sigma} \sum_{m=1}^{M} d_m(t) \frac{r - u_m}{|r - u_m|^3} \tag{6.21}$$

With parameter t as the time, a multiple dipole model for M dipoles $d_m(t)$, $m = 1...M$. Hereby, electrical potential $\phi(r,t)$ at position r for time t is calculated in the far-field, while σ is the conductivity constant and u_m is the origin of dipole m. In our model, the origin of a dipole is assumed as a constant, in contrast to the so called moving dipole, where the origin of the dipole is also a time varying function.

The time varying ECG signal $ECG(t)$ is defined by weighted potential differences. For an arbitrary body derivation[3] it is expressed, depending on the position of the - so called - different electrode r_d and $n = 1..N$ indifferent electrodes $r_i^{(n)}$.

$$ECG(r,t) = \frac{1}{4\pi\sigma} \sum_{m=1}^{M} d_m(t) \left[\frac{r_d - u_m}{|r_d - u_m|^3} - \frac{1}{N} \sum_{n=1}^{N} \frac{r_i^{(n)} - u_m}{\left| r_i^{(n)} - u_m \right|^3} \right] \tag{6.22}$$

Model parameters have been learned from the digital physiological signals recording database of the **PhysioNet** server (Physiobank) [183][83]. The parameters of a system for a multiple dipole model have to been specified, according to the potential measurements in the far-field. This process is known as the backward problem of the electrocardiography and can be expressed as the following optimization problem, if 12-channel ECG measured $data_i$, $i = 1..12$, are assumed as external potential differences for known positions of electrodes.

$$\sum_{i=1}^{12} \| Data_i(t) - ECG_i(t, d_m) \| \overset{d_m, u_m}{\rightarrow} min \tag{6.23}$$

The solution of this inverse problems, by the meaning of the dipole origins u_m and dipole description r_m can be solved by multivariate approximation techniques as Levenberg-Marquardt.

The solution of equation 6.23 describes a mixture of independent dynamic dipoles. Each

[3] Body derivations are standard electrical derivations with defined positions of sensor electrodes. Usually, one or many electrodes are taken as reference potential (indifferent electrodes) for a measurement on another electrode (different electrode).

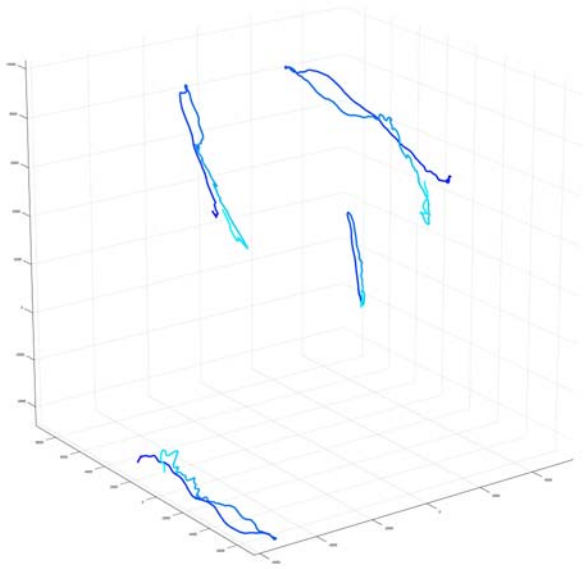


Figure 6.17: Trajectories of four different dipoles for a heart beat cycle, learned for the multiple dipole model, based on 12-channel real patient ECG data.

dipole describes a trajectory in the three-dimensional dipole space. The electro-physiological attributes are directly connected to this trajectories (e.g. to the time-dependent length and position of a dipole). Thus, it is important to provide a model-based representation for the dipole trajectories. Usually, a time-window of exactly one heart beat cycle is taken for this consideration. Sameni et al. [200] have presented a system of linear differential equations for this problem.

We provide a closed analytical model for this purpose, defined by equations 6.24. The model describes a Gaussian mixture for each dimension (d_x, d_y, d_z) of a dipole. The mixtures are defined on the circular orbit $\phi \in [0..2\pi]$ with the side condition $d(0) = d(2\pi)$ for each dimension. The latter condition describes the assumption, that the end of each heart cycle is defining the beginning of the next. One heart cycle is fully described by one circular orbit and each time point t has a corresponding position on this orbit, with appropriate values given by the Gaussian mixtures.

$$
\begin{aligned}
d_x(\phi) &= base_x + \sum_i^N xAx_i e^{\frac{-(\phi-\mu x_i)^2}{2*\sigma x_i^2}} \\
d_y(\phi) &= base_y + \sum_i^N yAy_i e^{\frac{-(\phi-\mu y_i)^2}{2*\sigma y_i^2}} \\
d_z(\phi) &= base_z + \sum_i^N zAz_i e^{\frac{-(\phi-\mu z_i)^2}{2*\sigma z_i^2}} \\
\phi &= \frac{2\cdot\pi}{T}\cdot t \in [0..2\pi] \\
d(0) &= d(2\pi) \\
t &\in [0..T]
\end{aligned}
\tag{6.24}
$$

The parameters $base_x, base_y, base_z$ herewith, describe general additive noise. This noise can include high frequency noise, which is typical for ECG measurements, as well as low frequency artifacts, given by movements of body or the regular chest movements due to the breathing process.

By the model defined in equations 6.24, we are capable to describe each heart cycle by few Gaussian functions and to model and simulate noise and artifacts as well. Typically, the Gaussian functions describe peaks or changes in the signal trajectory, which are corresponding to important electro-physiological phases, like P-, T- and U-waves or the QRS complex, as depicted in figure 6.18.

As one can see, by this model, it is possible to correlate the time position of each ECG specific wave (P, Q, R, S, T, U) to a parametric representation by a Gaussian function. By this significant process, we are able to support the modeling of additional electro-physiological important theorems, like QT time dependencies according to Bazzet or Firdercia [125] [5].

$$
\begin{aligned}
\mu x_Q - \mu x_T &= \Delta_{Bazzet} QT = \lambda_{Bazzet}\sqrt{T}2\pi \\
\mu x_Q - \mu x_T &= \Delta_{Friderica} QT = \lambda_{Friderica}\sqrt[3]{T}2\pi
\end{aligned}
\tag{6.25}
$$

T describes the length of one heart cycle, determined by the RR-interval[4]. According to Bazzet or Fridercia, the relation between the QT time (ΔQT) to the RR interval (T) is

[4]The RR-interval is the time from one QRS complex to the other. The R-peak is usually the highest measurable peak of the ECG signal.

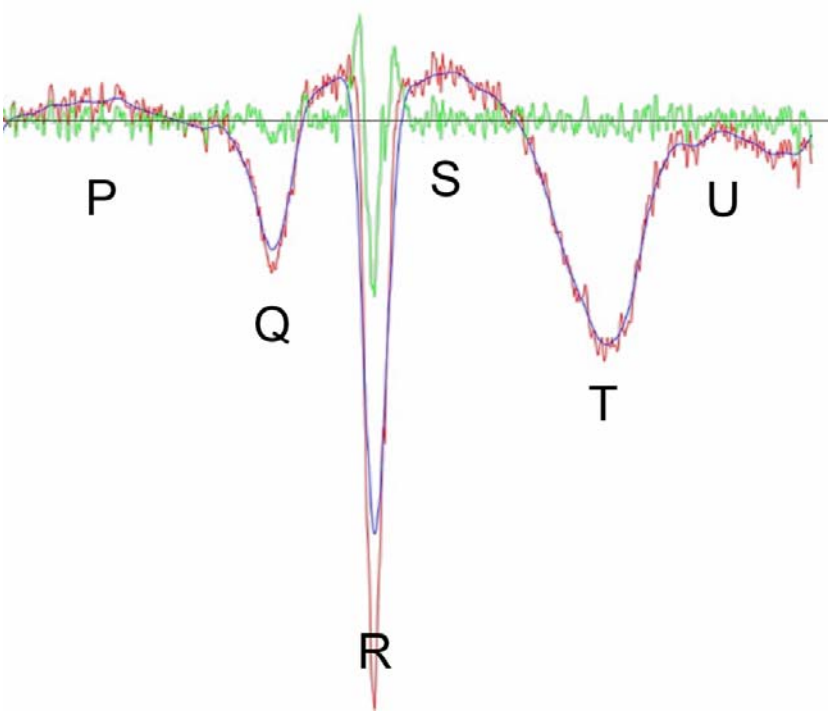


Figure 6.18: X dimension of an electrical dipole: the relation between model, noise and Q, R, S, T, U events is depicted. The red signal is the original dipole component. The blue one, represents the Gaussian mixture signal. The green signal shows the difference due to noise and model fitting errors.

constant (e.g. $\lambda_{Bazzet} = 0.42$) for a healthy heart at different heart rate frequencies. Changes of this relation for higher frequencies (e.g. under work load), is a diagnostic issue for potential heart diseases.

By the above model the following impacts were achieved :

- The model can be used as a diagnostic tool, correlating ECG signals by parametrized model descriptions, capable to measure QT time dependencies.
- One heart beat model can be used to generate many artificial (simulated) ECG signals with particular differences in signal shape, by varying QT time constants, HR or adding different artifacts (noise) signals.
- The ECG signal representation is very compact and reduced to 5-6 Gaussian functions in praxis.

In the following, an example of the XML model representation is given. The dipole models are described by the <sample> tag. Additionally, other simulation relevant tags are applied

to describe the emulation hardware, the resolution (sampling frequency), noise and other details.

```
<ECG>
<resolution>0.01</resolution>
<period> 0.5 </period>
<delta-period> 0.2 </delta-period>
<delta-noise> 0.04 </delta-noise>
<hw-interface type="12-channel">
  HW Interface
  <port>/dev/ttyUSB0</port>
</hw-interface>
<Bazzet> 0.42 </Bazzet>
<sample>
  <SD>
    <x>
      <A>0.12</A>
      <Phi>60</Phi>
      <My>25</My>
      <name>P</name>
    </x>
    <x>
      <A>-0.50</A>
      <Phi>150</Phi>
      <My>5</My>
      <name>Q</name>
    </x>
    <x>
      <A>1.2</A>
      <Phi>180</Phi>
      <My>5</My>
      <name>R</name>
    </x>
    <x>
      <A>-0.75</A>
      <Phi>210</Phi>
      <My>4</My>
      <name>S</name>
    </x>
    <x>
      <A>0.1</A>
      <Phi>300</Phi>
      <My>40</My>
      <name>T</name>
    </x>
    <y> ...  </y> <z> ... </z> ...
  </SD>
</sample>
</ECG>
```

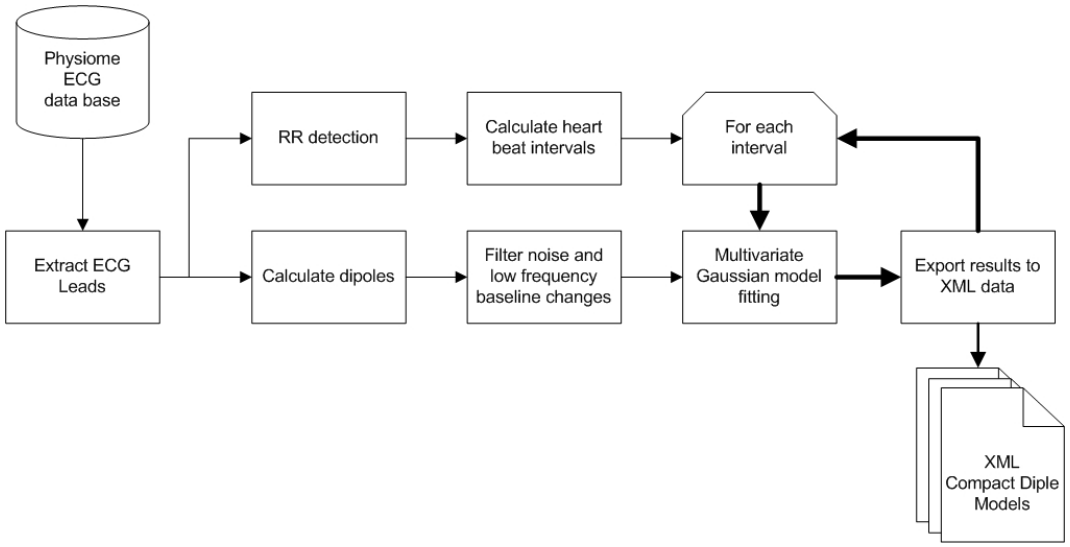


Figure 6.19: Dipole extraction and model fitting based on real patient data.

6.4.3 Learning From Real Data

The presented dynamic ECG model has many advantages. First, it allows to learn model parameter from real databases. Second, it provides a parametrized representation of the ECG time-varying signal dynamics, including knowledge about orientation of the dipole and electrical depolarization phases, e.g. the P- or T-wave and QRS-complex, which are usually clinical relevant, as well as their dynamic (time dependent) relation.

The task of learning from real data is pictured in figure 6.19, containing the complete process, beginning with the signal from the ECG databases and ending with a XML compact representation of the signal in form of dipole parameters. This task of learning has been implemented in Matlab. It is certainly interesting to optimize the parameter learning techniques for parameterizing (Gaussian mixture) of real patient ECG data in real-time for a better diagnostic reasoning.

The real data have been taken from ECG signal databases, provided by the Physionet archive of well-characterized digital recordings of physiologic signals (Physiobank) [83][183]. Based on this data, two independent threads are started. The first (upper) thread is responsible to extract heart beat intervals from the data, which is essentially based on RR detection information. The second (lower) thread is responsible for the calculation of the dipole information and preprocessing of the signal. This preprocessing includes various low and high frequency filter techniques.

For the high frequency noise removing a third order Savitzky-Golay FIR filter[5] has been applied, mainly because relative maxima, minima and width of the sample data are not flattened by this filter. The low frequency artifacts of the ECG signals are usually caused by movement of the chest, due to breathing. Many modern ECG monitoring devices use this information to calculate the breath frequency and automatically filter this information. The data on Physiobank, however, still contain the low frequency artifacts, which can easily be removed by a Butterworth [6] high-pass filter.

Once the preprocessing is completed, the dipole Gaussian model parameters (according to equation 6.24) can be calculated by regression analysis. The task is defined by formula 6.26, while d(t) is one multi-dimensional dipole:

$$\left\| d(t) - \sum_{i}^{N} A_i e^{\frac{-(\phi-\mu_i)^2}{2*\sigma_i^2}} \right\| \overset{\mu_i, A_i, \sigma_i}{\rightarrow} min \tag{6.26}$$

The minimization problem can be solved by multivariate analysis, e.g. least square optimization using gradient based methods, or by using expectation maximization techniques. In the implementation, the author decided to use polynomial multivariate regression techniques and calculate approximatively the Gaussian parameters according to the Taylor series development of the Gaussian function. One reason is that many regression analysis techniques (e.g. the mentioned ones) are very sensitive to initial conditions. This initial conditions can be defined very easily by polynomial approximations, which for this special case already match very good to local solutions.

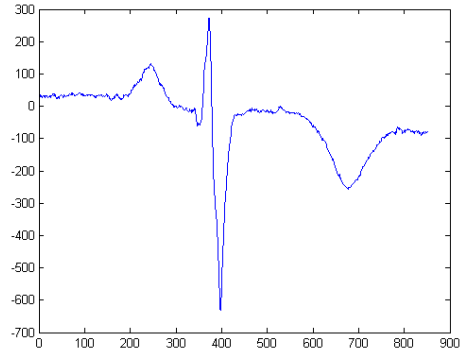
The local solutions are applied iteratively to find the global solution of equation 6.26, according to a significance level, given by the signal amplitude (using the baseline as reference). This is demonstrated in figure 6.4.3.

6.4.4 Characteristics And Results

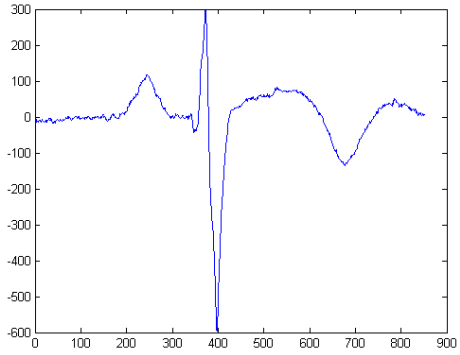
The client is a hardware interface, which supports the emulation of electrical potentials. For communication with the server USB is applied. The forward problem of ECG is solved on a μC realized on the client hardware.

[5] According to Abraham Savitzky and Marcel J. E. Golay, essentially describing a local polynomial regression, which removes higher order components.

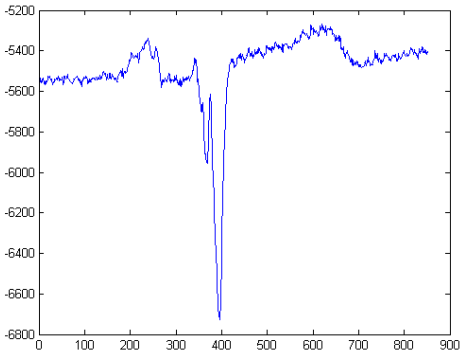
[6] According to Stephen Butterworth. This filter has a maximally flat magnitude frequency response beyond the cutoff frequency.



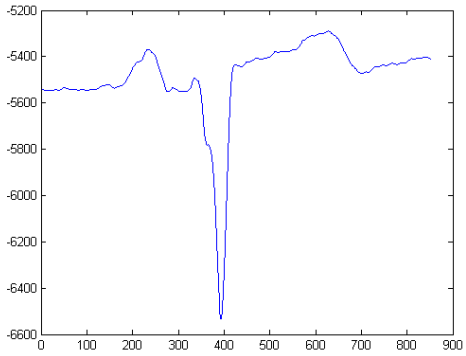
(a) V_x component of a dipole.



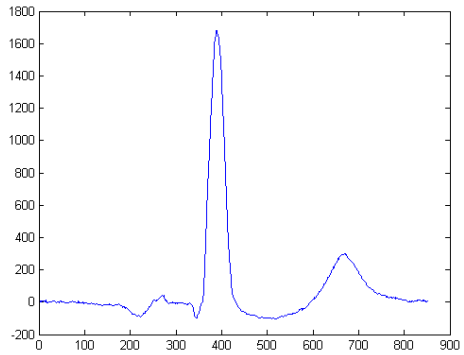
(b) Low frequency filtered V_x component.



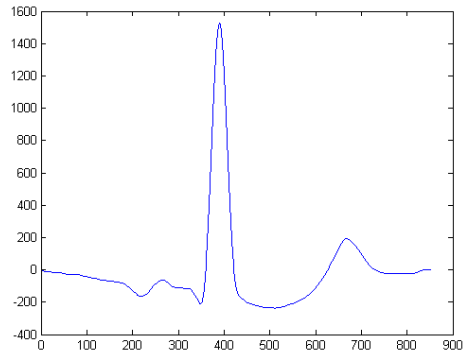
(c) V_y component of a dipole.



(d) High frequency filtered V_y component.

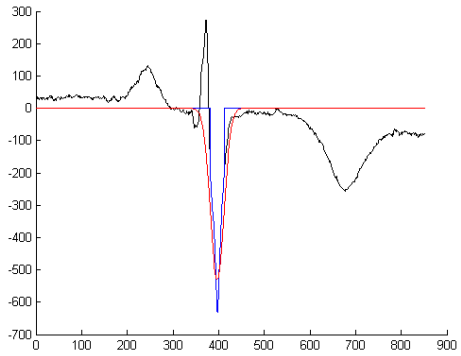


(e) V_z component of a dipole.

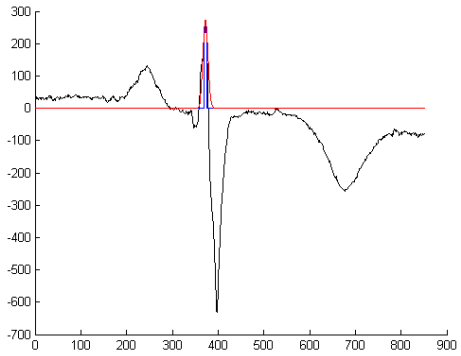


(f) High and low frequency filtered V_z component.

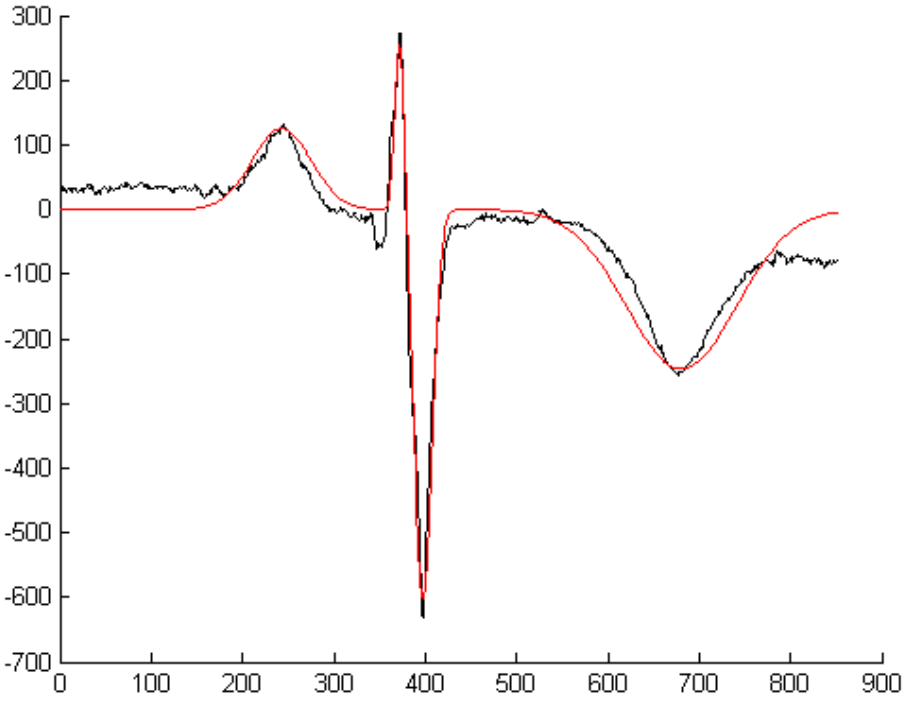
Figure 6.20: ECG model learning based on real data: First the dipoles are extracted from the real data for different heart beat intervals. The three-dimensional components of one dipole are shown in the figures (a, c, e). To remove noise and artifacts from the data, low and high frequency filtering is applied. A third order Savitzky-Golay FIR filter is used to remove high frequency noise (d, f) and a Butterworth filter to remove low frequency breathing artifacts.



(a) y-component of a dipole: Step 1 of the fitting algorithm.



(b) y-component of a dipole: Step 2 of the fitting algorithm.



(c) y-component of a dipole: Results after 4 steps of iterative model fitting.

Figure 6.21: ECG multivariate Gaussian fitting procedure. The iterative process is applied on the y-component of a dipole for one heart beat interval. The step by step process is depicted in (a) and (b). The approximative representation by a Gaussian mixture model with 4 Gaussian functions is already showing a very compact and good representation of the dynamic ECG signal.

In figure one, the custom designed ECG emulator 2-layered board is depicted. The PCB is supplied by power over USB (the connector is on the right side), the main DAC components

(on the bottom layer) are controlled by the micro controller MSP430 (top view in the middle). The emulated potentials can be accessed by connectors (pin-out at the left side).

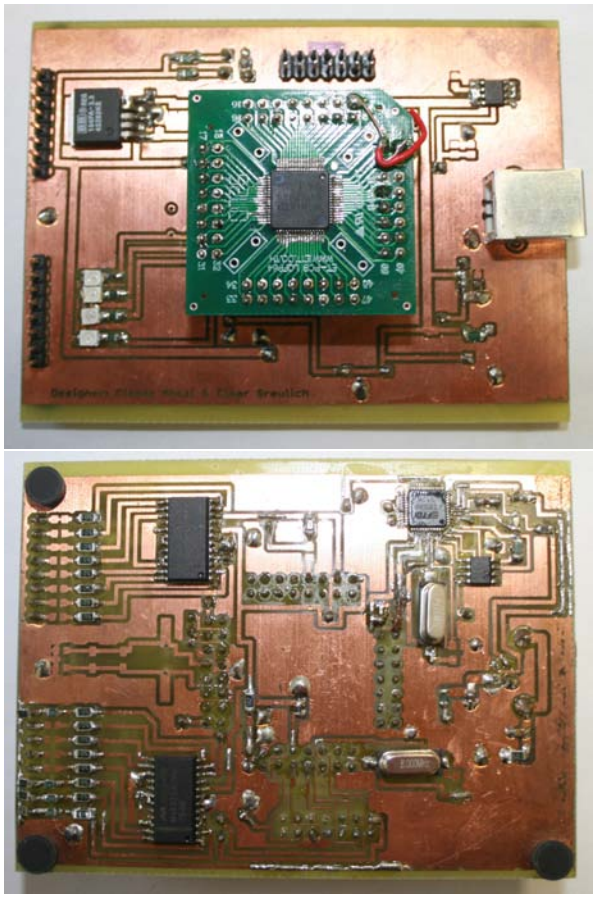


Figure 6.22: Top and bottom view of the ECG custom design emulator 2-layered PCB prototype.

The author has shown, that the single and multiple dipole(s) can be used for electrocardiography simulation and emulation (HW interface for real electrodes) in real-time applying the forward solutions of the electrocardiography in the far field.

Furthermore, the algebraic Gaussian mixture model allows a fast solution to parameterize the ECG waveform based on real-data, including noise, QT and QRS time dependencies.

The amount of data reduction is huge, since typically 5-6 Gaussian functions are sufficient to describe a full ECG cycle. By introducing the model parameters, it is possible to correlate physiological models to the ECG wave form simulation, which is used in our virtual human patient project. It still has to be analyzed how far a correlation to heart diseases is possible by the extension to multiple dipoles. In any way, with multiple sources more knowledge and waveforms can be represented in comparison to a single vector cardiogram.

7 Mixtures Of BN/DBN For Medical Simulations

One problem to develop BN and DBN models from real physiological data is to access real patient data, which is very complicated in EU and needs to pass special ethical issues. During this project, it was much easier to access psychological real data. Therefore, in the next section psychological simulations are considered, which are based on real data. After that, physiological simulations with BN and DBN are shown, which are based on simulated data and expert knowledge.

7.1 Psychological Simulations

Simulation of psychological effects are becoming more and more interesting in many areas of science. This topic is very related to the proposed simulation engine because of two fields of interest.

1. **Physiological Monitoring** Modern studies have shown, that the physiological monitoring systems, have a fundamental lack of information overloading [32] in a field where frequently stressful conditions could affect the appearance of human error [93], leading to patient critical situations. This outline the importance of adaptation of the system to the level of cognitive load or processing of the corresponding user. Although modern patient monitoring systems like the iAssist [16][32] have faced the problem of information overloading, the approaches here focus on context sensitive information preprocessing instead of adaptation or using information from the current user. New concepts however, focus on adaptation of the level of automation (adaptive automation) [34] to a user profile, which should be based on psychological and cognitive abilities. For the latter methods are necessary, allowing to reason such factors from user information e.g. based on user inputs from Human Machine Interaction (HMI) devices [120]. Psychological simulations,performed by the proposed system, have shown to be well-fitted for the above mentioned problems.

2. **Physiological and Medical Simulation** The role and importance of psychological simulation for medical simulations is becoming clear, considering the Maryland virtual patient. In 2009 cognitive models [206] with impact to patients health states have been integrated to this simulation [170]. For this simulator the patient's anamnesis and history is also regarded as a cognitive process, which is take into simulation [117]. The MVP system even consists of two different parts, a physiological agent and a cognitive agent, which form the virtual patient [155].

7.1.1 Psychological Assessment

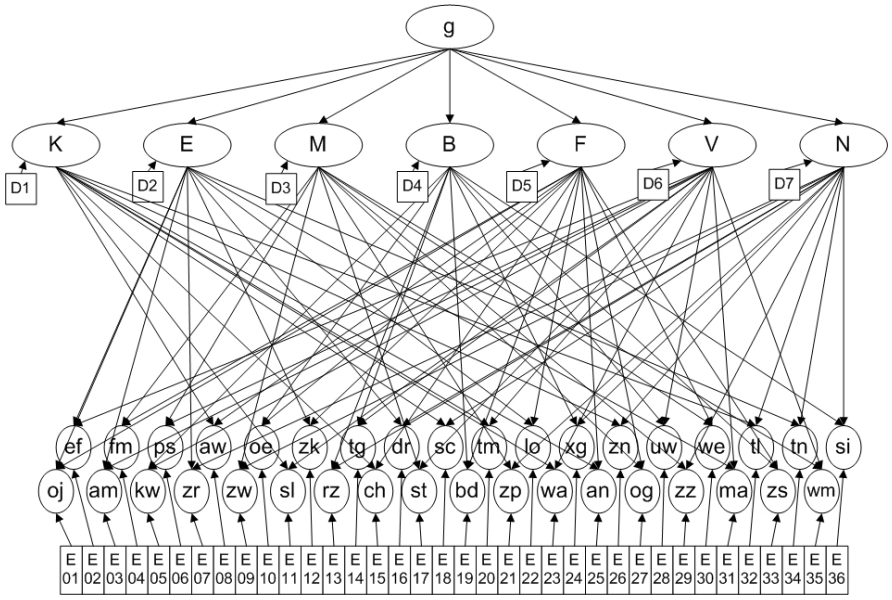


Figure 7.1: SEM applied for modeling and reasoning of cognitive abilities.

Up to now the most used method - considered as a standard approach - for the cognitive and psychological assessment is the SEM [180]. SEMs are applied in order to test and evaluate causal relationships between measured variables and factors/abilities, which can only be inferred indirectly via correlation and covariance matrices [120].

Typically cognitive abilities are modeled by using SEM, which is demonstrated in figure 7.1 on the example of a cognitive test model. The graphical network represents similar to a causal probabilistic network causal relations between different nodes, however additional nodes are applied to model data disturbances and error values (squared nodes). The top node "g" is here representing the general cognitive ability or cognitive intelligence, which is dependent to various sub-abilities called **cognitive/intelligence factors**, according to the definition of the classification of the Berlin Intelligence Structure Test [116].

The cognitive factors represented in figure 7.1 are among others measures for how good a person can process information in different ways depending on the content nature of information. Classifications are information processing based on auditory and verbal knowledge (node V), on figural and object knowledge (node F)or on numerical ones (node N). Another category describes the information processing and operations, including processing time (node B), retentivity (node M), ingenuity (node E) and processing capacity (node K).

To measure these cognitive factors and the resulting overall cognitive ability, up to 45 different psychological tests can be applied, while in praxis usually less tests are feasible (e.g. 36 psychic tests, represented by the bottom nodes of the SEM network). The parameters of the tests and cognitive factors are assumed to be disturbed by errors, usually described by Gaussian distributions.

Due to the parameter and structural uncertainty of the task, however, it is obvious to use approaches, capable to deal with model disturbances. Therefore in 2008 the author could demonstrate in cooperation with psychologist (Dr. Meike Jipp), a new method based on BNs, showing the usefulness to fulfill the task of psychological assessments in general [120]. The model was applied for practical problems, such that the gap between theory and practical implementation was closed. Details about the model as well as results will be shown in the following.

7.1.2 BN Based Modeling

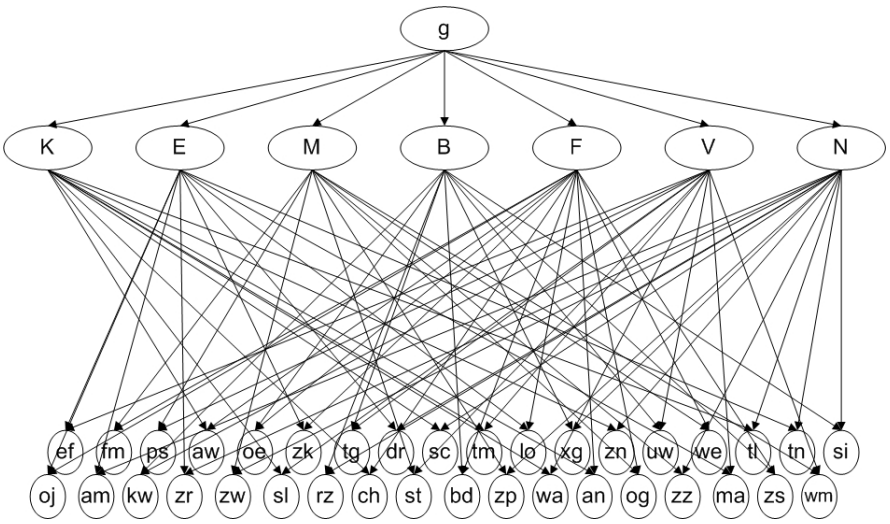


Figure 7.2: Hierarchical BN applied for modeling and reasoning of cognitive abilities.

The corresponding hierarchical BN model applied for reasoning and modeling of cognitive

abilities is depicted in figure 7.2. This structure is based on the knowledge from SEMs consisting of two layers. In the bottom layer the dependencies between tasks and intelligence factors and in the top layer between the latter and the general intelligence "g" is given by the structural connections. To not limit the degree of freedom of the characteristic CPF of each node as in the continuous case to a limited analytical distribution function (e.g.Gaussian), one choose different techniques for discretization (e.g. normal or equal distributed, equidistant or dynamic methods) as well as different number of discrete states to classify the data for a discrete BN analysis.

7.1.3 Results

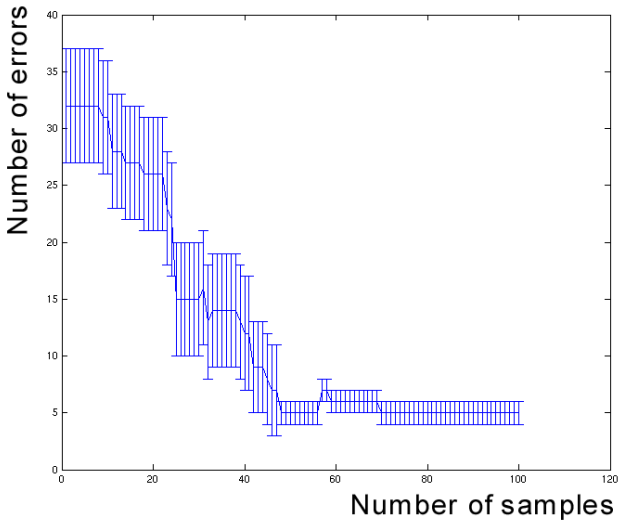


Figure 7.3: Parameter learning: The curve shows the learning effect of the network. 1-100 samples are used to learn the parameters of the BN and 33 are used for validation. The sample data are based on real user data.

The psychological test set - as defined before - is as well used to learn BNs as well as to validate the results. For parameter learning the improved EM algorithm [176] is used. In figure 7.3, one can see the parameter learning curve. On can see the relation between the total number of errors (Y-axis) in comparison to the number of learning samples (X-axis). 33 samples have been used to validate the network. If the network is not trained probably, the errors are maximized. The convergence is strongly dependent on the complexity, given by the number of discrete states. In this example the network is trained by 45 samples. The maximal discrete standard deviation is one discrete step, which is related to error in the network structure or the discretization.

The quality of modeling in terms of positive matches is depicted in figure 7.4. Here the network fit to the original 133 user data is visualized. As one can see, the network fits despite of maximal errors with size one perfectly to the original data.

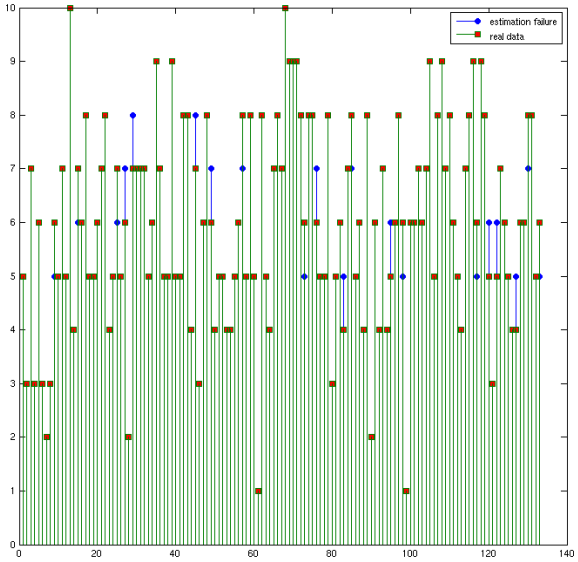


Figure 7.4: BN model matching: All 133 user data are used to validate the network fit. The maximal error is one discrete state. The total number of discrete states for each nodes was 10. This demonstrates the good fit of BN models to user data, in contrary to traditional methods.

In table 7.1 the runtime properties for the modeling and reasoning with BN for the above mentioned problem setting of cognitive assessment are shown. The measured time for parameter learning and reasoning is declared depending on the number of nodes (NN), number of discrete states for each node (NS), number of network layers (NL), and sample size applied for the parameter learning (SS).

Although no computational optimization techniques as GPU acceleration are applied, the time necessary for calculation of inference as wall as for parameter learning procedure are acceptable for real-time applications. Due to the fact, that the network structure was given by a single connected graph the computational complexity is linearly in number of nodes and their states, which is also reflected in the measurement times. As one can see the time for parameter learning is also very acceptable for online application, which means that the system could easily adapt itself to data or model changes during the runtime by renewing the model parameters applying online parameter learning.

			Parameter learning		Inference
NN	*NS*	*NL*	*Time[sec.]*	*SS*	*Time[sec.]*
8	5	1	0.046	100	0.012
8	10	1	0.047	130	0.012
44	5	2	0.083	133	0.063
44	20	2	0.108	133	0.063

Table 7.1: Runtime properties performed on a Pentium dual core 2.66 GHz standard PC. NN: number of nodes, NS: number of discrete states for each node, NL: number of layers, SS: sample size needed for learning

The model presented in figure 7.2, however provides a better result for reasoning than classical SEM approaches, however has still some disadvantages, since it is the classical way QMRs are modeled by connecting every node to every node and rising the network's complexity enormously.

A better way is to put more intelligence to the structure, which is especially in this domain very uncertain. Therefore various structure learning approaches were applied on the data and the results were surprising. By using PC and NPC [124] constrained based structural learning, one could adjust the structure of the network according to a Level of Significance (LOS), mirroring the data independency between nodes in the BN on the basis of sample data.

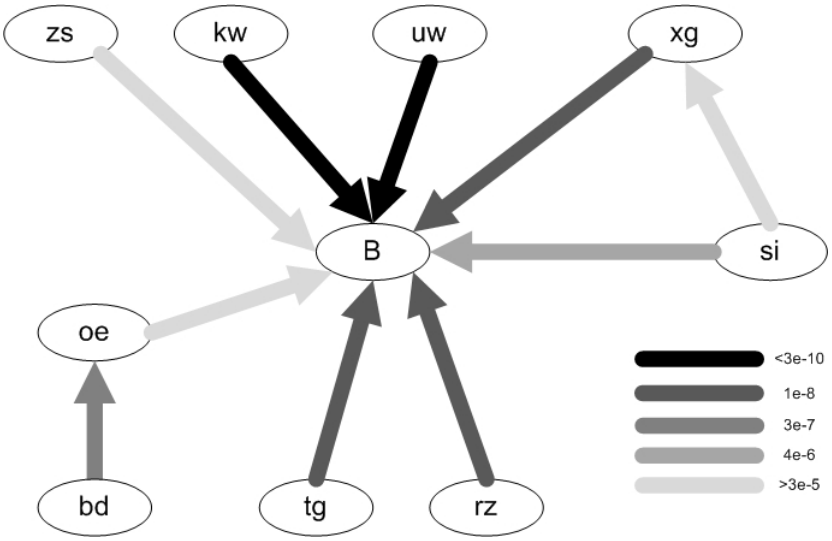


Figure 7.5: Structure learning : relations between cognitive tests and cognitive factors

However, due to the lack of clinical data and time psychological simulations based on BN could't yet be integrated into a fully patient simulation, by the applied techniques and methods two essential facts could be demonstrated. First, BNs are very applicable for modeling and

reasoning of psychological relations. Second the proposed simulation system proof the concept to show the applicability of real-time psychological simulations.

7.2 Physiological Simulations

7.2.1 Role Of BN

The role of BNs in medicine is underlined by various applications as well as future trends [149]. The main medical application is in the field of diagnostic networks (especially applied fir diagnostic reasoning). Most of this networks are not dynamic and are mainly used to fit data or knowledge to a probabilistic causal model. Good examples are the **heart disease network** [80] for description and diagnosis of heart failures, the **alarm network** [23] for critical situations during anesthesia and the **Hepar network** [238] for detection and diagnosis of liver disorders.

7.2.2 Role Of DBN

Although PBN are becoming important for medical reasoning and prognosis [175][237][236], DBNs have not yet found so much influence into modeling of physiological processes in clinical domain. One of the few applications of DBNs in medical domain are the modeling of genetic dynamic influences in 2003 [178] or the construction of genetic networks from micro arrays in 2008 [61]. Additionally the only comparable existing work on this domain is the modeling of glucose insulin regulation and simple cardiovascular processes by DBN in 2006 [106]. The results were formulated by a patent in 2009 [41].

However, this work was pioneering the application of DBN in medical domain and especially for modeling of physiological processes. Especially due to the lack of real data, many questions are still open. In contrary to the methods presented before, which show the impact for real user data. In the following, the results are based on simulated sampling data.

7.2.3 Comparison Of Methods

One significant advantage of DBNs for modeling dynamic systems is the option of using expert knowledge for developing the network structure. The latter is very important especially in medicine, where physicians have a deep insight in the matter. Without that knowledge, various

structure and parameter algorithms can be used to learn network structure without using prior expert knowledge.

So far, DBNs were used for modeling of health-disease processes without regard of underlying physiological phenomena [230]. This leads to correspondingly simple models, reduced to a general form, and the choice of amount and step size of discrete time steps is given statically by the disease progress.

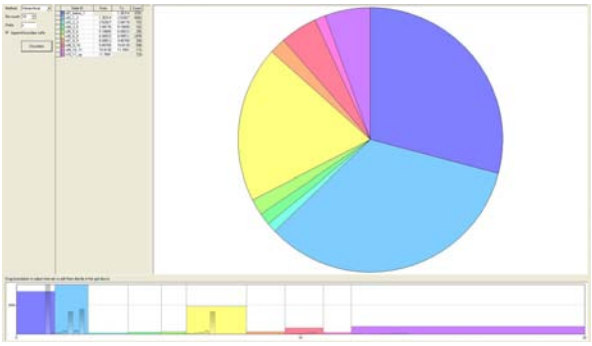
The most used model in literature to describe physiological processes is the compartment model. A compartment model is a representation for a system of ODE, which are used to express the dynamic changes of the system in reliance on prior system states and given parameters. By this way the dynamics of the system are described by mean values, representing a deterministic description. However one can show that for each linear system, that there is a representation given by causal networks existing [54]. This is essentially the reason why DBN can also be used for control of dynamic systems [60].

On the other hand most of the systems are linear and the descriptions and models used (e.g. ODEs) are simplifications and usually based on linearization of non-linear systems. Additionally factors like noise or uncertainty of parameters and model structure are usually neglected in such models too. Therefore a representation by probabilistic dynamic networks seems to offer better options to model complex, large and hybrid systems [142].

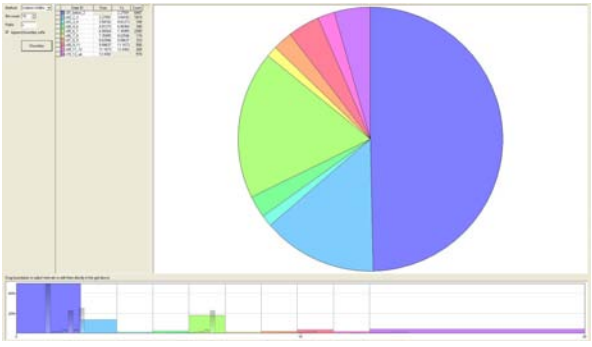
The structure and size of the network strongly depends on the number of discrete time slices and the number of discrete states of the network nodes. Additionally the choice of the discretization method influences the CPD and thus the structure as well. This is exemplary depicted in figure 7.2.3. Especially, if the density distribution of the samples is not normal distributed and one discretized under a wrong assumption, this will lead to a under-sampling of the network information.

By our general approach any compartment model can also be described by a DBN system, as shown on in Fig. 2 on the right side. Static dependencies are shown by normal arrows and dynamic dependencies by dotted ones. As one can see the knowledge about the transfer parameters (K_{21}, K_{12}) is hidden insight the learned parameters of the network, which is recommended if parameters are constant. If a parameter - e.g. clearance (O2) - is not constant it can be modeled by an anchor node with no dynamic arcs.

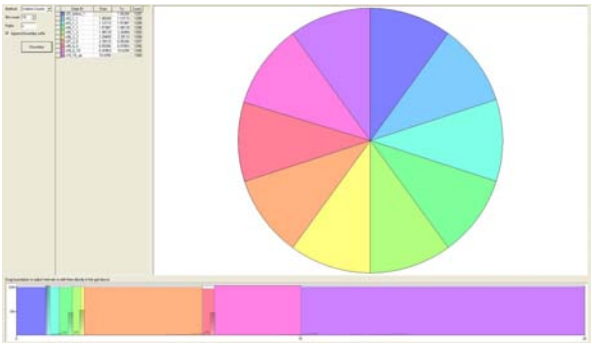
This leads to a simplification with regard to the number of parameters, which can be huge for complex physiological systems. But the main advantage of the proposed modeling is the expression of dynamics not only by terms of stochastic mean values of Gaussian but also for



(a) Hierarchical discretization is applied to fit the discrete distribution of the samples according to a normal distribution.



(b) Uniform length discretization is the easiest methods, but leads to unbalanced sample distribution.



(c) Uniform size discretization leads to an equal sample distribution.

Figure 7.6: Different discretization techniques: Genie GUI interface provides a good alternative for discretization of sampling data for BN learning purpose. One can see the impact of different methods on the distribution of samples to the discrete states.

any arbitrary probabilistic density function, without losing the strength of expression and simplicity given by the compartment models.

Assuming a temporal window of size two, simulation results for a 2-compartment model have been used to learn structure and parameter of a corresponding DBN. The result is depicted

in figure 7.7 on the right side, based on learning from 7000 sampling data and a discretization of the parameter space by using 10 discrete states for each node. The strategies and methods for parameter and structure learning are the same as discussed in the sub-chapter before, and therefore will not be exploited again.

On the one hand, reducing the number of states for each node below five, will lead to information loss and the network structure will degenerate. On the other hand, if the discretization is too fine granulated, the complexity will be too high as well (compare results in table 7.2).

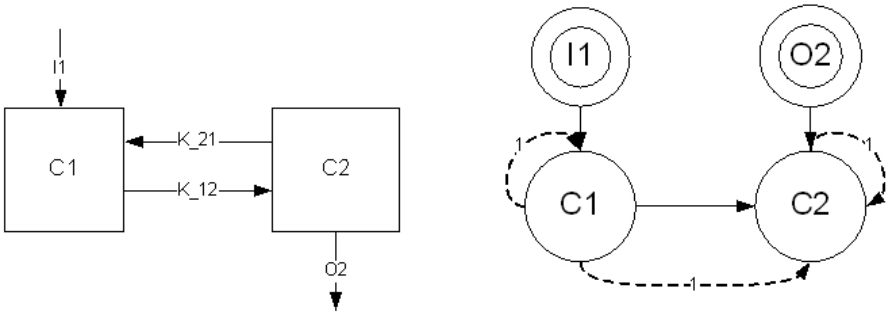


Figure 7.7: Comparison of DBN with Compartment models. In principle each compartment model can be represented by a corresponding DBN. In this example, the network was trained with 7000 sampling data and the temporal window has the size two.

To show the computational applicability in comparison to numerical integration methods (RongeKutta4), ODE based simulation results for a circulatory system have been used to learn a DBN model as well. Originally the circulatory system is described by 64 parameters. In the DBN, only the haemodynamic pressure information is used as representation of the dynamics of the system. Static values can be simulated by anchor nodes and will not affect the complexity substantially, but for simplicity are not considered in the following.

As one can see, the number of discrete steps per each node (which is for simplicity here the same for every node, but does not need to be) is fundamental for the computational complexity. However, 20 discrete classes to represent the circulatory pressure range (0-200 mmHg) is including the necessary clinical information. As can be seen in table 7.2, the computational sampling time is comparable with results for integrative models.

In our studies, we have shown that compartment simulations, e.g. of the circulatory system, can be used to learn structure and parameter of corresponding DBN models (e.g. by PC/NPC algorithm), leading to a compact knowledge representation. Due to the discretization, continuous physiological parameters are classified into corresponding discrete representations. Therefore, for fine-granulated simulations (regarding timing and signal space), DBN would miss their original aim. In our system DBN are therefore used for higher-level (low time

	ODE	DBN	DBN
Number of parameters	64 (8 Compartments)	8 (8 Nodes)	8 (8 Nodes)
Number of discrete states	-	20	50
Sampling time	2,8 ms	2,5 ms	16 ms

Table 7.2: Runtime properties performed on a Pentium dual core 2.66 GHz standard PC for comparison of DBN and Compartment models. Simulation results are given for a circulatory system. with different number of discrete states. The maximum depth of temporal arcs is 2.

granularity) simulations, which usually affect lower levels for physiological simulation and thus parameters like vital continuous vital signals. A good example is the DBN for ECG simulation, depicted in figure 7.8, which is developed according to expert knowledge.

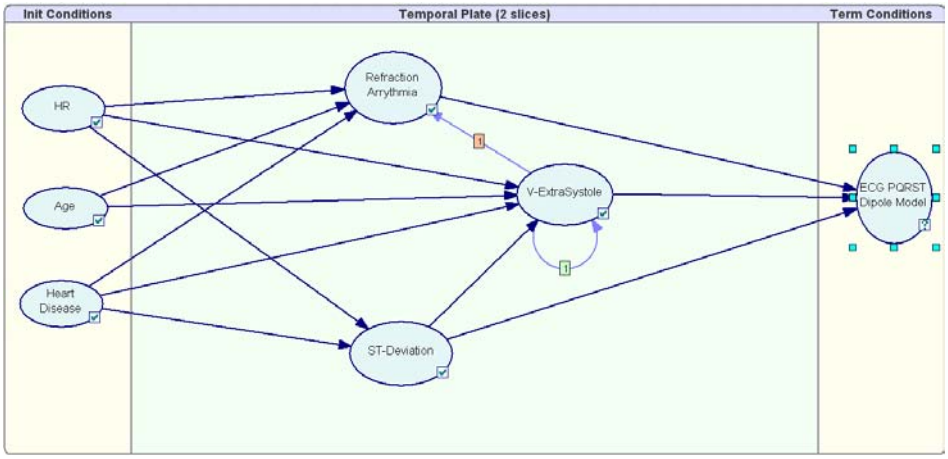


Figure 7.8: DBN model for ECG signal simulation: This higher level model, is sampled at each HR frequency. Anchor nodes, e.g. "Age", "HR", effect dynamic nodes, e.g "V-Extra Systole". By this, the effect between diseases and probabilistic knowledge about disease dynamics and the effects to the ECG signal shape (dipole model) is described, using a compact representation.

8 Other Simulations And Results

8.1 Integrated Simulation Environment

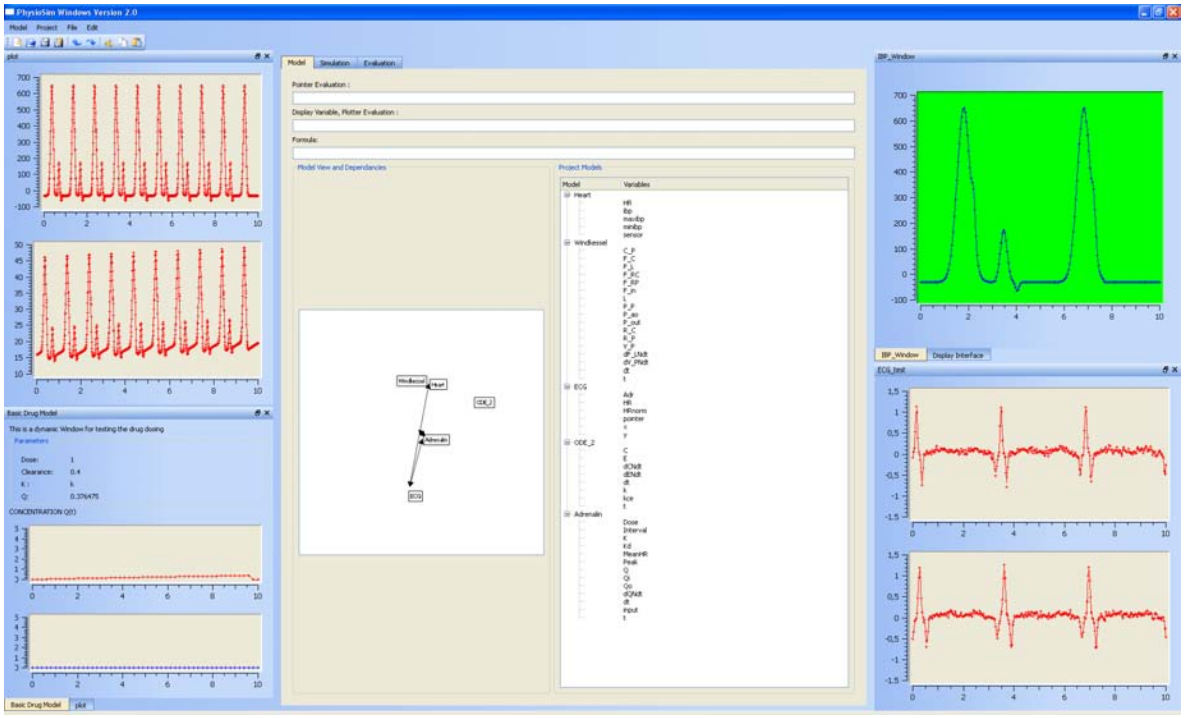


Figure 8.1: PhysioSim simulation environment (Software). The system is used to control real-time simulation process. One can load and unload physiological model components and manipulate model parameters.

The main result of this project is a medical simulation environment containing physical components (hardware) and logical ones (software), it demonstrates the feasibility of the proposed methods and architectures and will probably serve as basis for further research, analysis and developments[1].

[1] The environment is actively (Mai 2019) used for modeling of cerebral vasospasm processes and their impact to the cerebral haemodynamics. Additionally, an EU project is planed to integrate the simulator into a VR medical teaching environment, including real physiological reactions.

The software control GUI is depicted in figure 8.1. It is usually used to manage plug-in and out of different models into the system and to control variables and supervise model dependencies and structures. Model view and manipulation is supported as explained in the chapters before.

Originally, each simulation run-time model is described by a XML model description. More than 150 different models have been evaluated within this project, and to discuss each model will go beyond the scope of the work. Therefore, in the following sections partial models, results and simulation data will be presented, focusing on different strengths of the PhysioSim simulator.

8.2 Windkessel Aorta

We use a multi dimensional system to describe the windkessel vessel as a basic and generic sub-model for later developments of haemodynamic simulations. This model is a nucleus for more complex hemodynamic systems. The basic Windkessel model is given by the definitions in chapter Basics at page 34. The results of such a vessel to an artificial sinoidal flow input is depicted in figure 8.2.

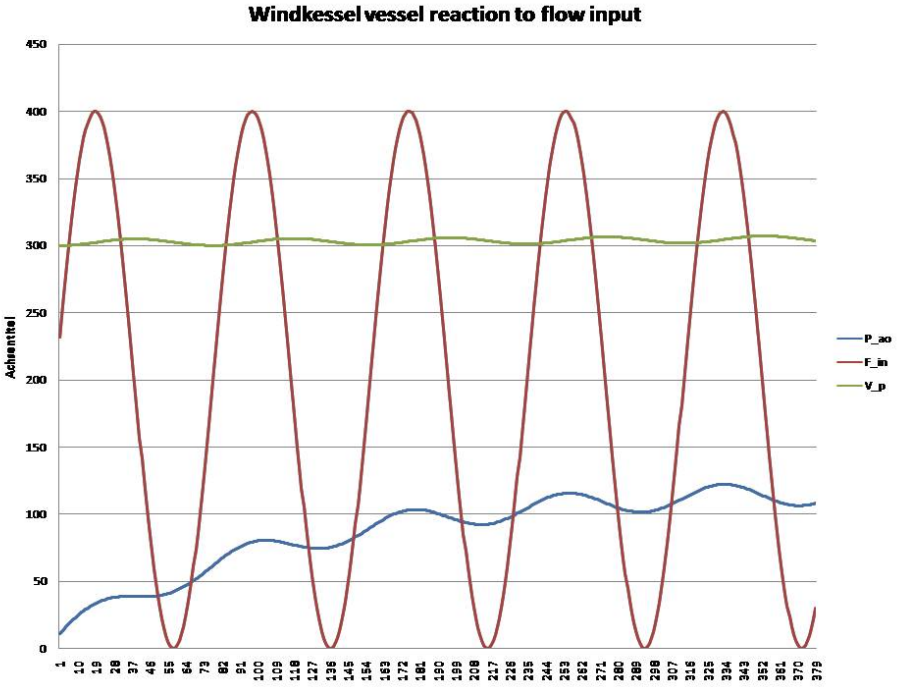


Figure 8.2: The windkessel effect on an artificial sinoidal flow input.

By using a realistic input flow e.g. given by the left ventricle ejection, one is able to go beyond the standard simulations, known from literature [215] and provide a very realistic simulation of the time varying blood pressure of the aorta. The simple system can be described by two sub-models as depicted in figure 8.3.

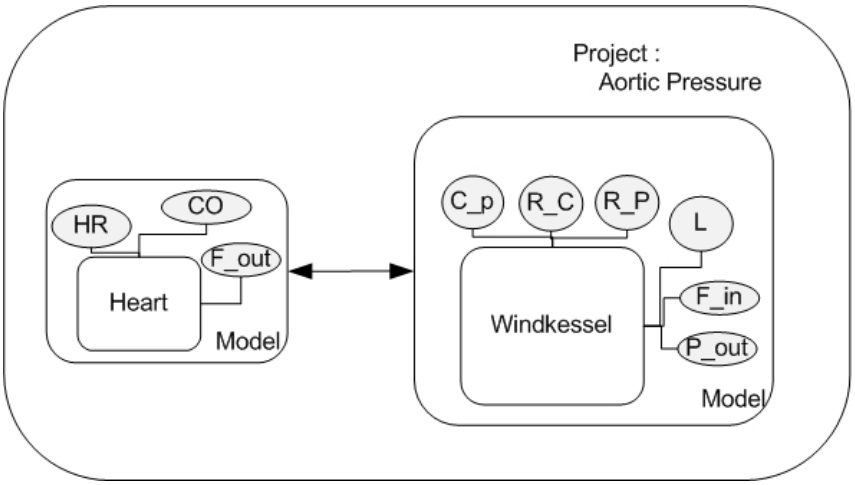


Figure 8.3: System to simulate realistic aortic pressure.

One heart sub-model is used, to describe the flow into the aorta based on cardiac output and heart rate variabilities. Depending on the elasticity and resistance of the vessel the aortic pressure is simulated in a very realistic way, as depicted in figure 8.4.

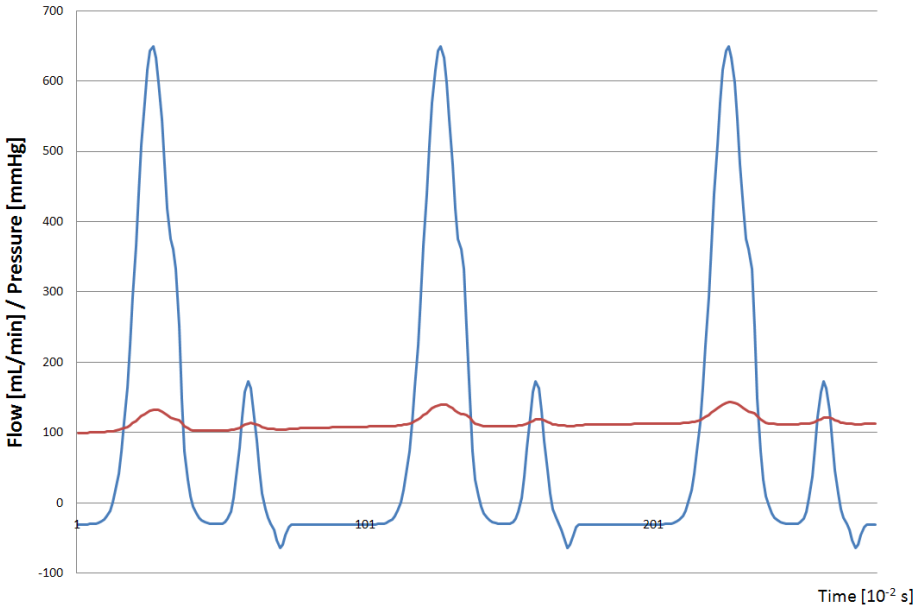


Figure 8.4: System to simulate realistic aortic pressure.

8.3 Haemodynamic Simulation

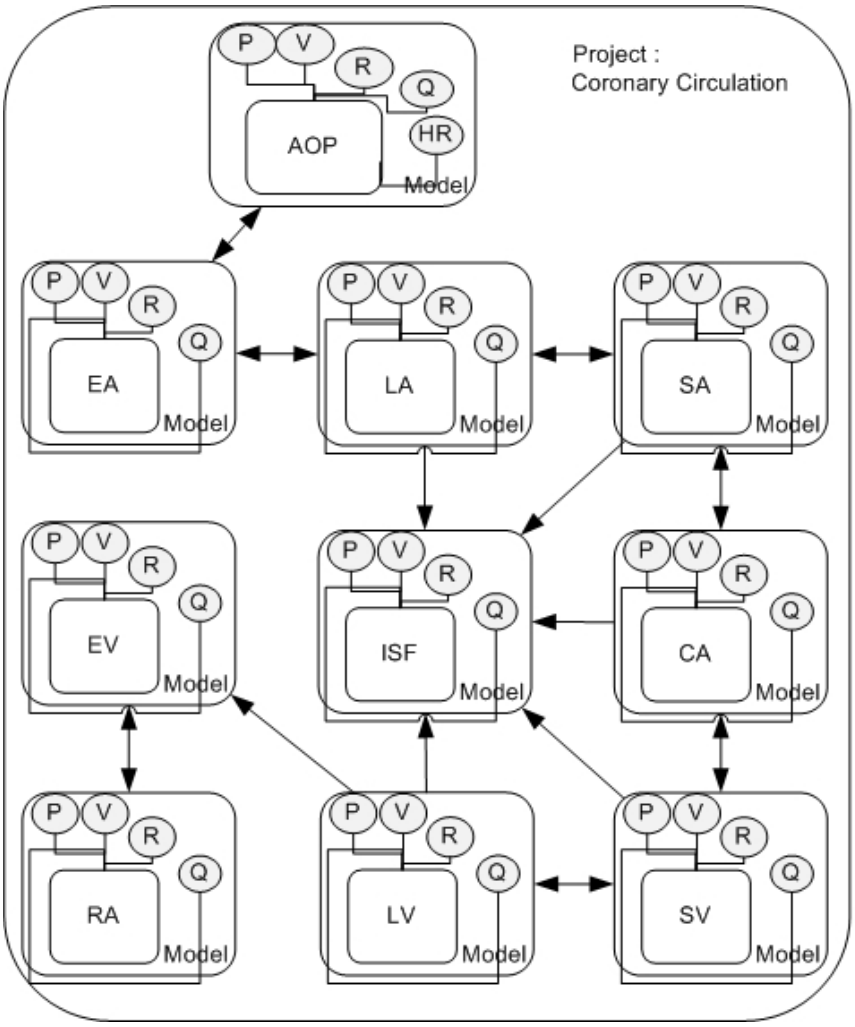


Figure 8.5: Haemodynamic model components - The following sub-models have been considered: aorta proximal (AOP), epicardial arteries (EA), large (LA) and small arteries (SA), capillaries (CA)R, small veins (SV), large veins (LV), epicardial veins (EV), right atrium (RA) and interstitial fluid (ISF).

One of the independent projects realized on the proposed system is the simulation of the haemodynamics a circulatory system. The complete model or project is composed by 10 compartmental component models (each described independently by a separated XML model), as depicted in figure 8.5.

The proposed methods are general and can be applied in various areas (e.g. for haemodynamic simulation of cerebral circulatory or coronary artery haemodynamics). We, hereby, restrict to a comprehensive model of the coronary circulatory system. This modeling is important

for training interventional treatments of coronary heart diseases with a vascular simulator. Especially pressure variations are important during the treatment for different locations on the coronary artery branches.

The simulation of coronary circulatory is a typical example for integrative system modeling and simulation. Typically, ten different compartments and 22 ordinary differential equations have to been solved for this simple model. More than 50 variables are to be considered here.

Simulation results are depicted in figure 8.6 and performed on an Intel Pentium 2.66 GHz within 2.8 ms sampling rate. One has to keep in mind, that not only the system of ODE has to be solved numerically but also the algebraic equations have to be interpreted by a math parser.

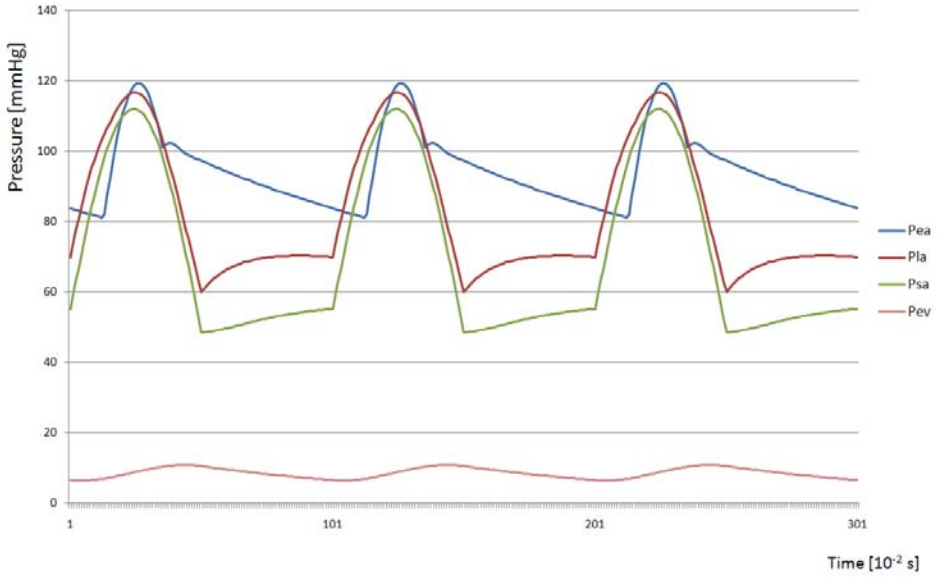


Figure 8.6: Simulation results for a circulatory system

8.4 PBPKPD For Atropine

Unfortunately PBPKPD modeling is not commonly used by Physiome or QCP projects, although the systems provide the necessary integrative mechanisms to describe such models. Therefore comparable models could only be found in literature. Usually the models can be found in the domain of pharmacology are very restrictive in parameters and do not use a standardized underlying physiological models, e.g. to describe the PK.

To demonstrate the applicability of our system, therefore the theoretical knowledge from

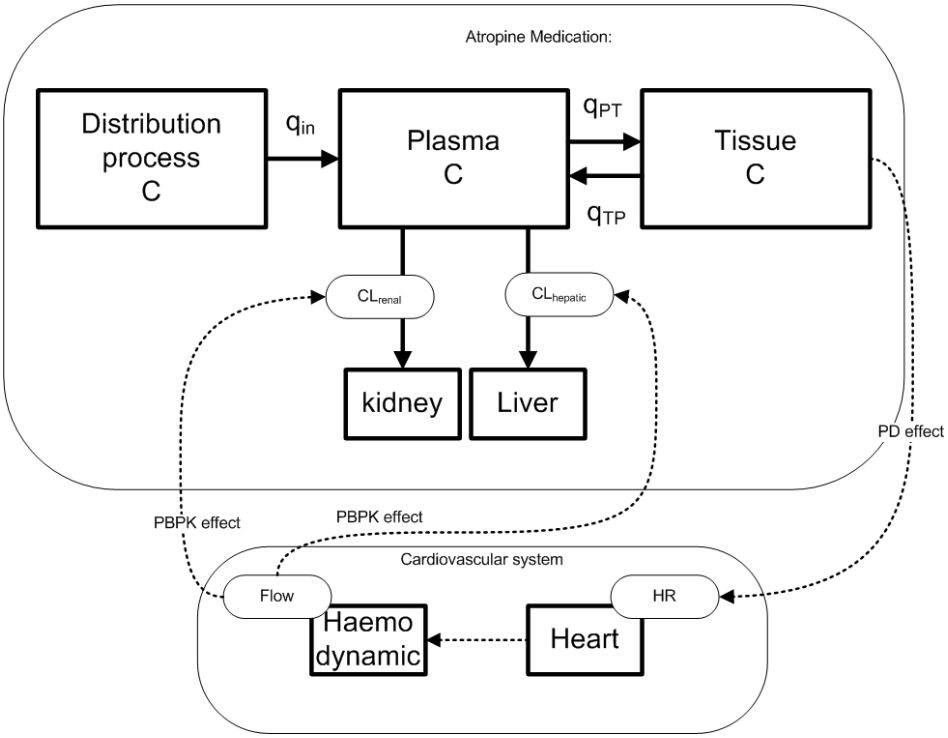


Figure 8.7: Atropine PBPKPD model

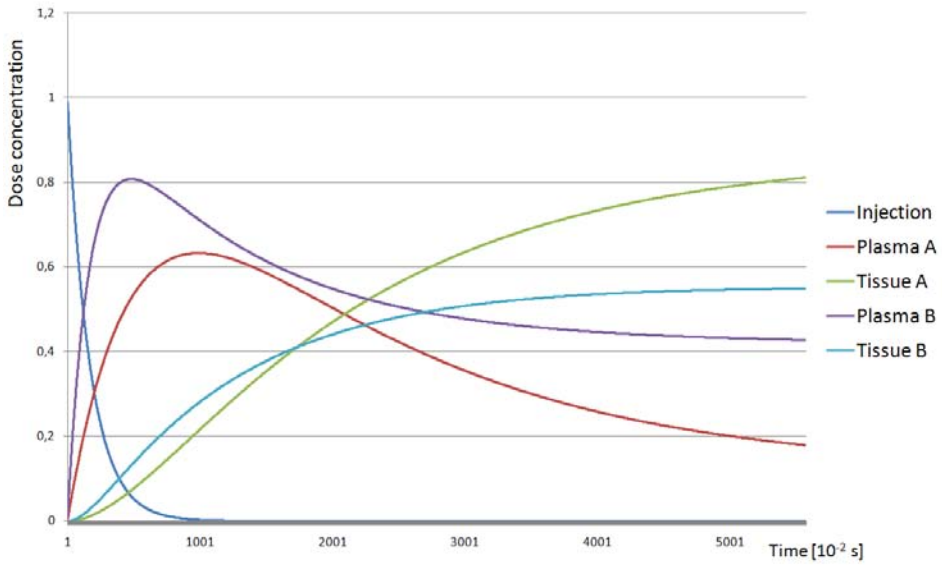


Figure 8.8: Atropine PK results: the dose concentration for a does of 2150μg is simulated according to clinical studies [102]. The plasma and tissue concentrations are depicted for two patients. Patient A has a renal extraction coefficient of 1.0 and patient B of 0.8.

literature was transformed exemplary to the XML model notation for an Atropine PBPKPD model. Atropine has especially, effects on the HR regulation mechanisms and is therefore used in therapy of various heart diseases or during cardiovascular interventions.

The physiological based model is based on compartment models for PK and PD, while the compartment parameters are dependent to underlying physiological parameters like flow [102]. The general modeling framework is presented in figure 8.7.

Results of the underlying PK simulation for two different patients A and B are shown in figure 8.8. Patient A has a renal extraction coefficient of 1.0 and patient B of 0.8. The injection dose was $2150\mu g$ intra-vascular. In the figure the total plasma concentration as well as the total tissue concentration are depicted.

The total tissue concentration is used to model the PD applying the modified Hill equations [102]. Typically, the PD effect is delayed in comparison to the dose injection and the dose plasma concentration, which is captured by this 3-Compartment model.

9 Towards Dependable Monitoring

One may wounder, what dependability has to do with physiological simulation? During this chapter, the author will introduce into the common problem of dependability, also for medical devices. A novel approach will be presented, how far physiological simulations can be used within this context. At the end of the chapter preliminary results will be presented.

9.1 Review Of Dependability Approaches

Reliability [20], security, Quality of Service (QoS) and dependability are measures (among many others) for the quality and thus for the trustworthiness of a system. Due to the close relation of these measures, they are often mistaken in literature. According to Avizienis et al. (2004) dependability is a hypernym including attributes like reliability, availability, safety, confidentiality, integrity, maintainability and other measures [18].

Typically a dependable system should prevent faults or deal with them in a confident way, in a way the user of the system is expecting. Therefore dependability is often related to the definition of faults, failures and errors. To access these measurement parameters usually methods or suggested, that in some way try to monitor and detect faulty situations. This process is known as monitoring of a system [142]. Usually stochastic methods are applied to model the mentioned attributes, e.g. Fault Tree or Success Tree modeling, stochastic Petri Nets, and BNs [20][240][142].

Although many approaches are established on this basis, most if them do either not consider the dynamic range of a system or have lots of difficulties to detect them if the description includes non-linear dynamic attributes. In the latter, which is usually the case for real systems, one tries to apply linearization techniques like applying extended Kalman filter instead of normal ones [142]. The fault tree for example is a static model and its nodes usually describe discrete faulty states of the system. To overcome this problem, Rüdiger et al. (2007) presented a new definition [197], to quantitatively compute dependability measures for dynamic systems [198].

Due to the growing complexity of medical devices, particularly defining safety critical applications such as the health-care information and monitoring systems, risk analysis for clinical monitoring is becoming increasingly important [152]. The efforts on this field are focusing on the reliability analysis e.g. by fault-tree or BN [169] modeling for the technical system and do not take into account the errors which are related to the human operator or user of technical systems [120]. New approaches, therefore, not only model the technical system, but also the interaction of the user or operator in terms of modeling the human. These approaches show a huge improvement of the systems, especially due to most of the failures of a system are related to human errors [121].

The request to integrate the above mentioned ideas into a better patient monitoring although present, could not be satisfied yet. Already 1993 Coiera [44] pointed out the problems of information overloading and cognitive mismatch in clinical monitoring. Also the weakness of static information extraction of current monitoring systems in comparison to the complex and dynamic real-time changes of the patient itself were clearly scrutinized [44] in "More intelligent monitoring and control of dynamic physiological systems in Medicine".

In PK and PD modeling this idea has been used to provide a better risk assessment already, because by the better models (PBPKPD models) including partial physiological considerations, a better medication planning is possible [68].

However, there are systems trying to particularly include the mentioned ideas, like "iAssist" [32] 2007, which is focusing on expert system development including more intelligent real-time algorithms for a better interpretation of physiological signals [63]. Currently no known system is applying a dynamic dependability analysis of the patient itself to improve patient monitoring. An idea, which is logically the next step [44] in research and is recently motivated by methods from dynamic system analysis [198].

Therefore in the next subsections a novel methodology is presented, which is focusing on system analysis, based on a behavior based description of the human patients physiology, allowing to access measures for the patients dependability by taking the patient into the system modeling, as part of the system itself. This can be, in the opinion of the author, a basis for a new way of thinking for the design of intelligent patient monitoring systems.

9.2 Dynamic Definition Of Dependability

For human physiology, however the internal system states are very difficult to find, due to the amount of uncertainty to describe the overall complex system. However the vital signals

are important signs and measures of the patient internal health states and usually directly connected [201]. Additionally, these signals are usually observable by patient monitoring systems applying medical sensors. Therefore during the next pages a method for patients health dependability measurement is suggested, which is based on measured signals, especially vital signals like ECG, IBP and SaO2. Signals, which are mentioned in previous chapters about simulation. Indeed, simulation of such physiological vital signals will play and important role for the suggested definitions, that will follow now.

For this purpose we assume that the space S is describing the multi dimensional signal space of different signals over time t. The signal $S_i(t) \in S$ is then describing the time trajectory of $i-th$ signal in this space.

In figure 9.1 a typical HR signal trajectory is depicted. One can see the fine-dotted trajectory, which is describing the predicted/computed changes of the patients HR due to a medication over time, according to results based on simulations, e.g. run at PhysioSim simulation environment.

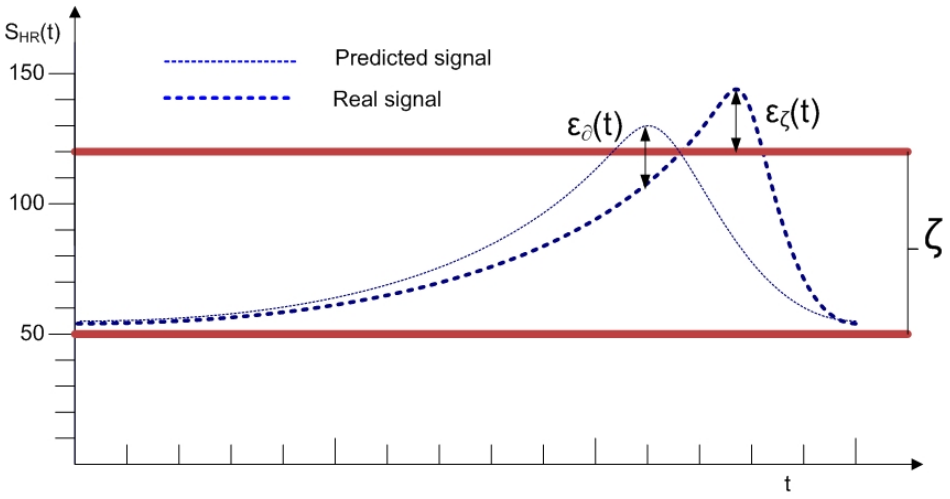


Figure 9.1: Signal trajectory and signal space S exemplary for HR changes over time. Two HR time varying signals are depicted. The fine dotted signal is a simulated signal, according to physiological reactions to a medication. The bold dotted signal is the real/measured patient HR signal trajectory. The red thick lines are representing static safety boundaries $\zeta \subseteq S$, typically for patient monitoring systems.

The red lines are marking static safety boundaries $\zeta \subseteq S$. Safety boundaries are typically in dependability analysis, because by them faulty situations can be defined. In patient monitoring safety boundaries are also known as **alarm** boundaries, because usually warnings and alarms (in terms of acoustical and optical signals) are connected to them to warn the medical personal and prevent life critical situations.

However, alarms and warnings are pointing to a patient's health critical situation. Typically for HR one is expecting a minimum HR of 50 *beats/minute* and the maximum of the range is about 120 *beats/min*. However, this definition has nothing patient individual and is a very static statement. Therefore, in praxis usually alarms of such system are disabled, because they do not fit to an individual patient. The reader may wounder, but exactly by this, their overall purpose to prevent patient critical situation, is missed.

Although in literature and praxis requested a lot to change the structure and basis of such monitor devices systematically [32], in clinical daily routine more intelligent patient monitoring systems are unfortunately still far away from use. One first step would surely be therefore to change from static safety boundaries to dynamic one.

According to Rüdiger et al. (2007) [197] safety boundaries should not be static and time independent, because the boundary may change according to the system's dynamics. If the dynamics of the system are given by a behavior based description of the system, then the dependability of the signal trajectory for a time window t_w can be specified as a function of time:

$$D_{t_w,\zeta_i} = 1 - \frac{1}{t_w}\int_{t_0}^{t_0+t_w} \epsilon_{\zeta_i}(\tau)d\tau \tag{9.1}$$

while ϵ_{ζ_i} is given by the Euclidean distance:

$$\epsilon_{\zeta_i} = \|S_i(t) - \zeta_i(t)\|^2 \tag{9.2}$$

for every signal trajectory $S_i(t)$ and the corresponding safety boundary trajectory $\zeta_i(t)$. The Euclidean distance, herewith, is a measure of the deviation from a dynamic safety boundary corresponding to the meaning of an error.

This definition of dependability is thus related to a time window t_w under consideration. If this window is small, recent errors (defined by equation 9.2) will play a more important role, for large time windows, they will be smoothed due to normalization.

This formalism has two main impacts for future patient monitoring systems, on the one hand the boundary $\zeta_i(t)$ can be any time varying arbitrary trajectory, which could be adapted to the current patient. On the other hand the integrative window shows how the boundary error is behaving over time and thus past errors are taken into account as well.

Usually, system dependability is defined with respect to a special mission [198], which is usually defining timing boundaries as well. In our case, stabilizing a patient's health state by an intervention or a medication is describing exactly such a mission and corresponding mission trajectories. For such a case, we cam define another measure, the mission dependability regarding to mission boundaries, given by equation 9.3.

$$D_{t_m,\zeta_i} = 1 - \frac{1}{t_m} \int_{t_0}^{t_0+t_m} \epsilon_{\zeta_i}(\tau) d\tau \tag{9.3}$$

t_m is describing the mission time which is given by the time for an intervention or a medication or any other operation to change patient's health states. ϵ_{ζ_i} is again the quadratic error $\epsilon_{\zeta_i} = \|S_i(t) - \zeta_i(t)\|^2$, which is given by the Euclidean distance of the real signal value $S_i(t)$ and the signal safety boundaries $\zeta_i(t)$.

Hereby, the above definition is very adaptive and can also include the known safety boundaries, which are although static, actual state of the art in patient monitoring. However it is more interesting to have dynamic safety boundaries, which could be adjusted according to knowledge from expert systems or real-time simulations.

Considering the signal trajectories in figure 9.1, one can see that simulated/predicted signal trajectory (fine dotted signal) is showing what we expect the patient's signal to behave during a mission. The real signal (bold dotted signal) difference $\epsilon_\sigma(t)$ to this expectation is thus a measure for a irregularity, that can be caused by errors in the simulation model or errors in patients health state. This idea, leads to a novel approach of patient dependability monitoring, which will be presented in the next section.

9.3 Simulation Approach

However, the behavior based definition of dependability with regard to changing safety boundaries is obviously an improvement, standardizing and extending the terminology of existing alarms, which represent static boundaries, the major impact of a system theoretical approach is to predict future states and vital signal trajectories.

In a behavior based dependability approach, therefore the difference between forcasted / predicted (expected) and real signal trajectory is playing an important role. This measure is corresponding to the ad hoc definition of dependability[18]: a dependable system should react in a way, how it is expected to react and thus a dependably patient should.

In the previous definitions dependability analysis was based on time trajectory measurements dependability on a specific mission or task, under consideration. For dependable patient monitoring the measurement should also be related to a mission. Usually this is given by the therapy methods applied on the patient. Therefore the terminology of an **event** is introduced in this work, specifying time (start time t_0, mission time t_m) and way of therapy e.g. medication (specific physiological event model).

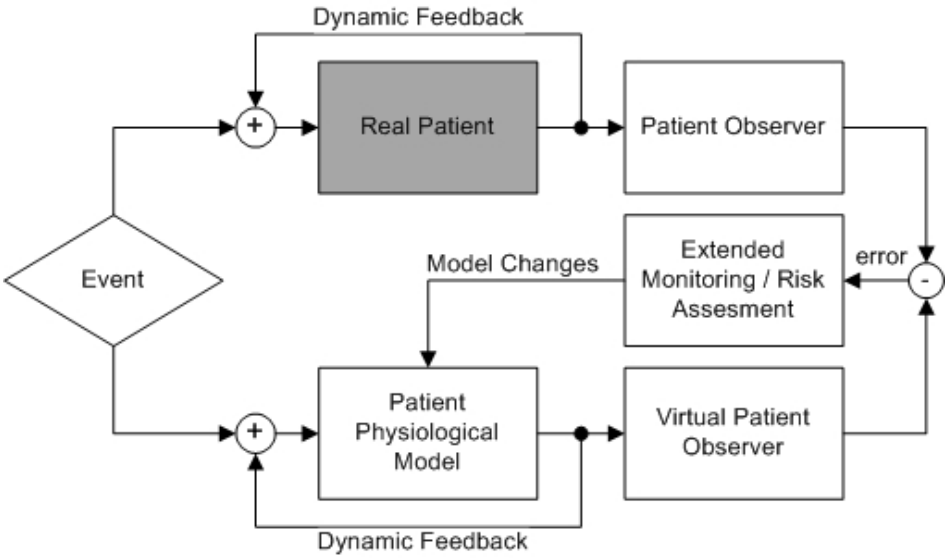


Figure 9.2: Vital signal observation and comparison between real and simulated physiological reactions to an induced event. A system theoretical view for patient monitoring is depicted. Upper part : real patient loop and observation. Lower part : simulated patient loop and observation. Intermediate part : comparator and extended dependability and monitoring analysis.

Figure 9.2 shows a system theoretic way of view for the given problem, namely to monitor patients physiological system by the meaning of patient vital parameters.

The upper part of the diagram shows a real patient block, being a black box model and including some observable and non-observable internal states. This block describes the physiology (behavior) of the patient, in other words the patient's health states, which could be multi parametric. According to system dynamics - subsequent patient states are correlated to earlier ones - a dynamic feedback loop is necessary. As mentioned before, we are unable to observe and measure all patient internal parameters, which is depicted by a patient observer block.

The lower part of the diagram is describing a the virtual physiological model, which can be realized by simulation. The result is a simulated patient block. Also here the model is a dynamic model, thus a dynamic feedback is needed and additionally results of the simulation

can be observed by an observer block. The patient model may be any mixture of time-invariant dynamic models, even containing non-stationary probabilistic temporal models. As proposed by this work a mixture of deterministic and probabilistic modeling approaches by the meaning of system of ODEs and BN/DBN mixture can be used.

If the virtual model is mimicking/simulating the real world perfectly, there will be no difference in both observations. A difference, however, is interpreted as error given by the simulation, which - as depicted in the intermediate layer - allows extending the monitoring by providing more knowledge about patient states and even extend to patient dependability and risk analysis. Normally, if the virtual patient model is accurate and well suited, the error is a significant sign for a deviation between real patient states and virtual patient states. Such a deviation may be interpreted as a deviation from safety boundaries and hints towards possible safety critical situations.

By the formalism of the event based simulation the start time t_0 is defined. And the simulation results as well as the real values for actual time $t \geq t_0$ as well as simulation results in terms of predictions for future time slices $t + t_w \geq t$ are known.

Then an event based dependability formula can be defined by eq. 9.4:

$$D_{event,i} = 1 - \frac{1}{t}\int_{t_0}^{t} \epsilon_{\sigma_i}(\tau)d\tau - \int_{t}^{t+t_w} \epsilon_{\zeta_i}(\tau)d\tau \tag{9.4}$$

while ϵ_{ζ_i} is given by equation 9.2 and ϵ_{σ_i} is given by :

$$\epsilon_{\sigma_i} = \|S_{i,simulated}(t) - S_{i,real}(t)\|^2 \tag{9.5}$$

The event based dependability measure $D_{event,i}$ is a measure for normalized integrated difference between real ($S_{i,real}(t)$) and simulated ($S_{i,simulated}(t)$) signal trajectories extracted in the intermediate layer of the system, depicted at figure 9.2.

The first part of this measure is considering past signal trajectories according to the definition of equation 9.3, describing how good the patient's signals fit to the expected signals. The second part is considering future signal predictions and their distance to specified safety boundaries $\zeta_i(t)$ from equation 9.1. Hereby, only the window of interest t_w is considered, describing a prediction/simulation for future signal trajectories, which is essentially dependent on how many predictions can be made by the system. Usually the more predictions are possible the less accurate they will become.

The error formula for the past can be interpreted on one hand as a measure for the quality of the simulation model. If the model is not simulating the real world accurately the error is large and the model is not well suited. By adding additional knowledge e.g. changing model parameters one adapts the model to the real world. This is either realized by user interaction or by applying multivariate optimization techniques. On the other hand if the model is designed well for healthy patients. The error term for the past is thus a good measure for the health state of a patient, taking time-variant information into account as well. Deviations to the health state is considered as reduced dependability like in system theory.

In our architecture, as shown in figure 9.2, we assume that there is a model which simulates and predicts the dynamic time-invariant changes of a monitored signals. Generally, accurate models are rare, because one needs to know the trajectory of the system states as well as the environmental influences. Therefore usually Kalman filters for linear systems and extended Kalman filters for nonlinear are applied as standard approaches for prediction of future trajectories. In general, probabilistic models e.h. HMM or DBNs are typically used to allow prediction of future system (in our case patient) states.

9.4 Quality of Service

In the architecture, proposed in figure 9.2, there is a possibility depicted to change model parameter according to time dependent information from real patients signal trajectories. This process of updating internal states of the dynamic virtual patient model by the knowledge of the real world observation is known as **smoothing** for probabilistic dynamic systems [71]. But not only parameter updates for probabilistic models also for deterministic ones can lead to another prognosis for the next prognosis time window horizon $t_w \geq t$ within a predictive model.

Assuming that we can apply N_w updates on the patient model within the time window t_w will result in a measure for the quality of the predictions for future outcome, as shown in equation 9.6.

$$Q = 1 - \frac{1}{t_w N_w} \sum_{i=1}^{N_w} \int_t^{t+t_w} \epsilon_\sigma^{(i)}(\tau) d\tau \tag{9.6}$$

Q is a measure of the quality for the predictions by values in $[0, 1]$. The error $\epsilon_\sigma^{(i)}$ is given by the quadratic Euclidean distance of the real signal value $S_{real}(t)$ and the predicted/simulated

value $S^{(i)}_{simulated}(t)$ at time t for $i = 1..N_w$ updates (model smoothing) within the prediction horizon:

$$\epsilon^{(i)}_{\sigma} = \left\| S^{(i)}_{simulated}(t) - S_{real}(t) \right\|^2 \tag{9.7}$$

Therefore $\epsilon^{(i)}_{\sigma}$ is dependent to the time distance of future time slice simulations (predictions). Additionally one has to consider that the entropy for probabilistic inference and thus the amount of uncertainty is increasing with the amount of reasoning steps N_w and the prediction time t_w [174]. The formula 9.6 is exactly confirming the law of entropy and will lead automatically to worst QoS for the predictive model, the more predictions N_w are necessary within the time horizon t_w.

The QoS formula is nevertheless a measure for the past predictions of the simulated model. By assuming model linearity this can be regarded as an extrapolation of future prediction quality. The value is smoothed over the time window for the prediction horizon, but could easily be extended to larger windows of interest.

9.5 Applicability and Interpretation

Our system developed for real-time physiological simulations is using a hybrid approach applying ODEs and DBN for simulation of physiological interactions[6]. It is based on a hierarchical model description such that basic models for circulatory can be connected with e.g. models for drug interaction or interventional models as well. This system has been used to show the feasibility of the suggested approaches in a central monitoring environment. Essential is thus to provide interventional models, e.g. for medication, such that an event based simulation as introduced for equation 9.4 is applicable.

We prepared a setup for a virtual ICU monitoring environment, as one can see in figure 9.3. A simulator dummy is used here instead of a real patient. The dynamics of the simulated patient are represented by a set of models (e.g. circulatory system, medication, respiration defined in a XML model library as also presented in chapter 8 (Results) and patient specific parameters. The quasi-real patient signals are collected by a standard multi parameter patient monitor (Tianrong TR910 [225]) and used together with virtual simulation results for dependability analysis in a central monitoring software system.

A similar simulation model is running virtually (no emulation is necessary here) on the central

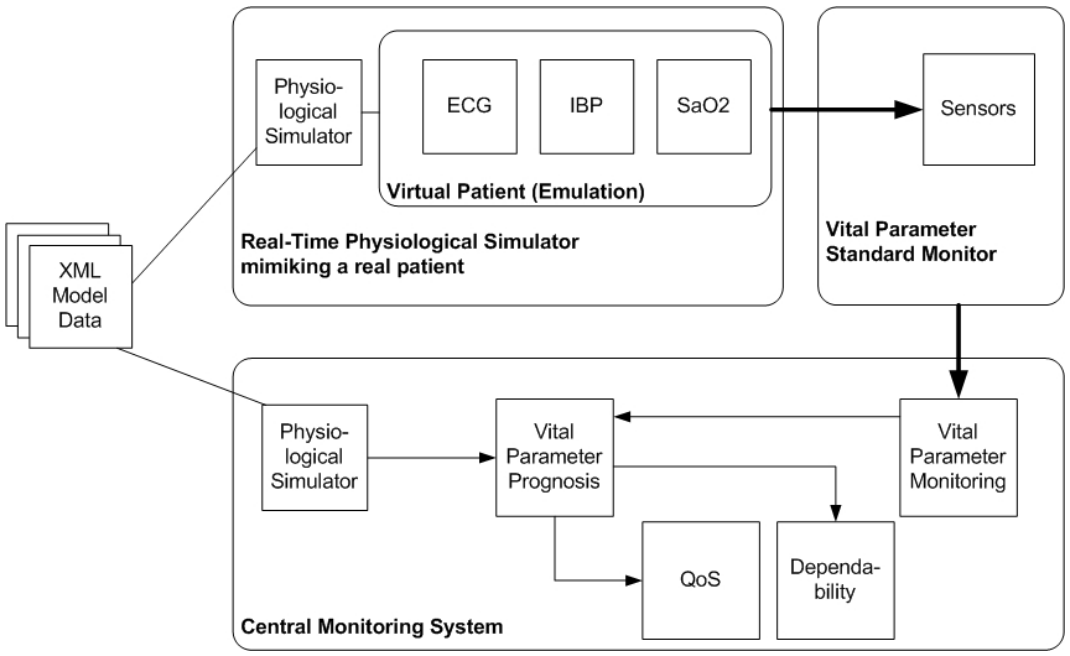


Figure 9.3: Test system for dependability analysis. XML Model data are the basis for a virtual and emulated physiological simulation. The emulated simulator is representing a real patient. Vital signals are collected by a standard multi parameter patient monitor. These quasi-real patient signals and virtual signals are processed for dependability analysis in a central monitoring software system

monitoring system, while here the model parameters could be others, e.g. those of a healthy patient. The virtual model updates internal states due to real measurements, emulated by the simulator dummy. The model prognosis is analyzed regarding quality of service as well as dependability aspects for risk assessment.

Exemplary a case study, simulating the effects of epinephrine medication was used to show the feasibility of the previously mentioned methods. Usually epinephrine is used to treat bardycardia. Therefore the focus was on the effects of this medication to HR changes. The results are depicted in figure 9.4. On the one hand a simulation (basic circulatory system in combination with simple 3-compartment PBPK) is running to forecast a prognosis for the effects on the heart rate (HR), on the other hand a similar simulation is running on the physiological simulator dummy to simulate the vital parameter in real-time. The measured data are processed by a monitoring system and emulate real data, although they are not from real patients. The error between forecasted and real data is used to compute the dependability value for the HR, given by the induced medication event. In fact, the error here is due to different parameter (clearance factor) given by the patient physiological model.

Applying dependability analysis on the human patient leads to interesting new methods for

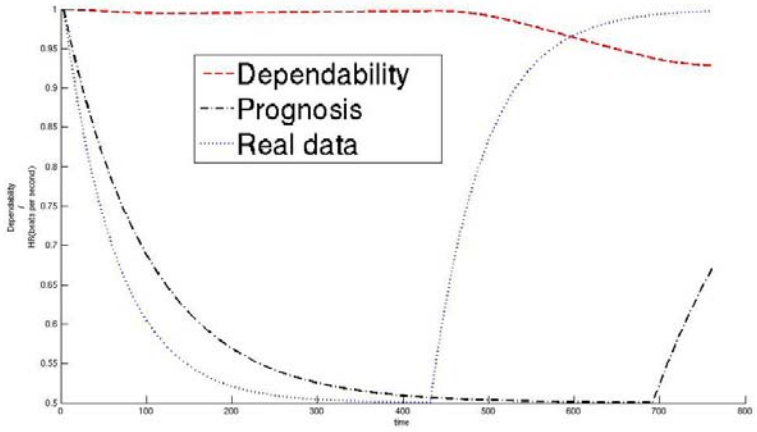


Figure 9.4: Dependability value based on simulation. Exemplary results from simulation of effects of epinephrine to HR are shown. One can see the forecasted HR due to the medication (Prognosis) and the real data extracted from the monitoring system. The error leads to a decreasing dependability value.

clinical monitoring. Physiological simulations are playing a key role in the proposed architecture, as far as they are addressed to take into account patient individual parameters as well as model updating and reasoning abilities. Once such models are available, the reasoning of events as medication or intervention for a specific patient based on the monitoring of vital parameter and other knowledge e.g. history, age and gender can be used for an individual risk assessment.

By our methodology, we have shown that there is a general framework to access the dependability of patient states without forcing fault-tree modeling or similar approaches known from the reliability/dependability analysis. On the one hand the dependability measure for future risk and past model differences is a new view on patient's critical situations; on the other hand the quality of service is a measure for the applicability of the virtual physiological models.

We are preparing in vivo experiments on rats to test our methodology for vital parameter monitoring based on dedicational injection, showing how such a system can be used to develop better and more specific models for drug interactions. As our approach is designed for general dynamic systems, future work will integrate these results to probabilistic dynamic systems to analyze the effect of uncertainty on the patient risk assessment. Static and dynamic BNs can support the processes of dependability analysis in a new continuous way, varying form the current state of the art methods of reasoning probabilities of faulty states (fault-tree like modeling with BN) as known from reliability analysis [136].

10 Discussion

10.1 Main Contributions To The Field

A full physiological simulator environment - PhysioSim - in hardware and software has been realized, emphasizing new methods and features in medical simulation technology. Physic and model-based physiological simulations were not considered in this detailed way for usage in medical simulator systems before. A hardware and software architecture is presented, supporting real-time physiological simulation and emulation.

Although the main focus was on simulator design and development, the PhysioSim system is fully comparable with software developments, systems and methods in physiological modeling, e.g. given by the European Physiome project (JSim) [182].

The key, in the opinion of the author, is to effectively combine and connect the two different research areas: theoretical research in modeling of physiological process, and the practical oriented research in modeling of physiology, e.g. for educational simulators. By this, not only medical educational simulators can be improved substantially according to the actual research, but also other domains as medical patient monitoring, or diagnostic expert systems.

Several new methods and models have been integrated into the PhysioSim system, enabling interdisciplinary research, providing interesting scientific results beyond the original task of simulator development, for example results in psychological modeling and simulation or in patient monitoring system development.

The thesis is thus addressing many different contributions to the scientific community, which will be discussed in next section (Contributions To The Fields). In the following, the advantages and features of the PhysioSim system in comparison to other systems will be discussed briefly:

1. **Hierarchical Modeling**: By the new approach of hierarchical modeling and model management, based on object oriented realization of models and model managers, it is

much more comfortable to design complex and huge models of the physiology e.g. as demonstrated for the haemodynamic simulation.

2. **Dynamic Model Management** : Due to the object oriented and dynamic model management, it is possible to load and unload models during the runtime. From the system theoretical point of view this process is related to system changes, which are usually non-linear processes.

3. **Integration of DBN/BN Mixture**: Other known approaches for physiological modeling have not yet integrated static and dynamic BNs into the modeling procedure. In this work, the author show the benefits of such extension clearly. Although static BN have been used for simulation of static probabilist causal relations, e.g. for simulation of complications in medical treatments, by integration of dynamic BNs additional degrees of freedom for modeling is offered.

4. **Integration of hardware interfaces**: As studies have shown, today's educational simulators are still far away from fully virtual and screen-based simulators, because the educational and training results are much higher for those simulators, with close relation to realism, fundamentally given by a realistic mimicking of a real patient using a patient dummy (e.g. the HPS). This underlines the importance of adequate hardware devices, capable to simulate and emulate patient signals. In this work three different novel autonomous hardware emulators for vital parameters were successfully introduced. A general framework was suggested to integrate such devices in full-scale simulators and into a physiological simulation framework, which was demonstrated as well.

5. **Real-time Simulation**: Essential for applying physiological simulation for medical educational simulators is the fact of providing real-time capable simulations, which is the greatest lack of other approaches. By using modern object-oriented programming techniques in C++ and continuous optimization of the source code, the system was capable to fulfill this requirement.

Furthermore based on the above system the following methodologies have been presented and evaluated :

1. **Modeling and reasoning of cognitive abilities**: A new method to model individual cognitive abilities was presented, which were originally described by SEM. Hierarchical BNs have been used for this purpose, dealing with uncertainty and noisy data. The validity of the method was demonstrated on the basis of simulation of real life data. The inference from the BN demonstrates that the current user's cognitive ability can

be demonstrated in real-time reliably. Structural and data dependency analysis show furthermore that less evidence data are necessary in comparison to traditional SEM to provide a cognitive profile classification, highlighting the advantages of the new method. A similar approach could also be applied to assess psychomotor abilities. Both results were published recently by the author [120] [121].

2. **Dependability analysis based on medical simulation**: Applying real-time physiological simulations in the domain of patient monitoring together with novel approaches in dependability analysis, lead to a new view of patient dependability analysis, which were presented in chapter 9. The revolutionary idea in the new approach is to consider the patient himself as part of a system, one wants to be more dependable. System critical states herewith are patient critical health states. The definition is very general and dynamic, extending the static view realized at the most current state patient monitoring systems. Results were published recently by the author [8] [7].

In summary, the following contributions to the different fields of research could be achieved:

- Physiological Modeling and Simulation Environments: Hierarchical modeling, hybrid modeling with deterministic and probabilistic models, dynamic object oriented model management, are novel strategies, which provide various new advantages and possibilities for physiological modeling and simulation.

- Hardware Development for Physiological Simulators: A generic architecture as well as three novel concrete hardware simulation and emulation interfaces have been presented. Focusing on modularity and extensibility of current simulator hardware.

- Software Development for Physiological Simulators: A simulator software architecture has been presented in detail. Important aspects are dynamic model management and hierarchical modeling based on XML. In contrary to script based simulators, model-based physiology can be described with this system.

- Physiological Models: Special purpose models, e.g. for ECG have been presented as well as generic hierarchical approaches for haemodynamic modeling. Event based modeling is introduced, by this work. An event like medication or intervention is fully described by a corresponding model, which can be loaded to the system on demand.

- Hybrid Dynamic System Modeling: By the provided methods, one is able to combine deterministic modeling approaches with probabilistic ones, which opens various new aspects for general dynamic system modeling. Especially DBN as a generalization of HMM are very promising for similar system modeling approaches.

- Cognitive Modeling: By BN and DBN modeling and simulation, there have been provided a new standard for cognitive reasoning and corresponding applications.

- Patient Monitoring: As shown by a feasibitliy analysis, simulation of physiology is highly correlated to patient monitoring applications. Corresponding models and theories have been presented, demonstrating the specific weaknesses of current non dynamic monitoring systems.

10.2 Open Questions

To formulate it with similar words as Scott Meyers [160], one experts a perfect system has to provide all the functionalities of a big system and has to be adaptive and extensible, while it should be as well minimal, or it should be hardware optimized and the same time platform independent. Such a perfect system does not exist yet and thus PhysioSim is not, although it has adapted more effective techniques to be more efficient. However, this - talking about efficiency - will always imply boundary conditions, which make a system development specific.

By focusing on physiological simulations for medical education, we have defied this boundary conditions, where specific aspects have been analyzed, developed and optimized with certainly slim time and man-power. Nevertheless, in the following, we will discuss aspects, which may be important outside the mentioned boundary conditions, e.g. for other scientific application with similar aim or aspects, which may need additional consideration.

10.2.1 Platform Independency And Specificity

By using C++ as basis for the software realization, one had the benefits of a controllable calculation time and could thus realize a real-time system, but the inevitably related disadvantage here can be regarded as the less platform independency. Physiome or other JAVA based physiological simulation tools can run on any platform, which supports a JAVA virtual machine. The proposed system in this work, needs to be compiled for each different system separately.

Nevertheless, nowadays the C++ libraries support many different platforms and due to the advantages of managed build and compilation processes, using CMake or QMake, a code can easily be compiled on different platforms. Our simulation system has successfully be compiled

and run on Windows XP Professional and Gentoo Linux system. Therefore, we assume that other platforms, will also be supported, may be after slight modifications.

Considering the aspect, that due to the growing complexity in number of parameters and system variables of physiological models, one has to focus on optimization techniques to accelerated the computing time for the physiological simulation environments, one can think of GPU based computational acceleration. To integrate appropriate techniques, certainly a C++ basis will be a better choice of the programming language.

10.2.2 Data Feed In

In this work various approaches have been proposed, which could help to provide better expert and assist systems for intelligent patient monitoring based on physiological simulations. However to provide realistic simulations, more complex models and validation with real data are mandatory. Furthermore, many physiological processes are still very complex, highly non-linear and are affected with lots of uncertainty.

Therefore BNs and DBNs can provide an interesting alternative to model such processes, if learning data sets are available. Of course such data sets are not static themselves and do usually grow and change over time. Therefore an important aspect will be the data feed from real data sources to machine learning applications. To talk about automation of this process may be is very ambitious at this state of the art, but will probably become more important in the next years.

For our simulation models, this process of model learning was done manually for static and dynamic BN. Then one has three different possibilities for model building. First, can apply only manual model design, thinking of expert knowledge or medical scenarios, which can be easily and logically be mapped by BN. Second, one can apply structural and parameter learning procedures. And third, one can combine the first two strategies by using constraint-based learning methods. For all methods, we have faced various advantages and disadvantages.

Using only expert knowledge can provide an effective way for modeling of scenarios for medical education, but is sometimes far away from the real-data. For instance, during cognitive modeling with BN, sometime the real-data show significant correlations, which were not yet considered by experts.

Learning from real-data on the other hand, can provide very compact knowledge representations, which are very effective and useful for medical simulation and modeling. But only

trusting on real-data learned probabilistic networks could also lead to huge modeling failures. This can occur if the sampling data or not statistical independent, are not well distributed over the multidimensional probability space, leading to under sampling (shadows) or oversampling (blinding) during the learning processes.

However, the combination of above approaches, using constrains-based optimization processes usually is an iterative process, where the designer can control the particular important aspects e.g. sampling, stochastic dependency, discretization and expert knowledge in the best way, according to the opinion of the author, also here wrong constrains (based on wrong expert assumptions) can lead to wrong models.

As one can see, this is a very sensitive area, where probably further developments can be expected in the next years and decays. Nevertheless, the possibilities and boundaries for the specific task of medical simulations have been analyzed and offer a basis for constructive research.

10.2.3 Dual Space Theory

In this work the dynamic systems, usually partially described by linear differential systems, were solved in time domain by numerical differential integration techniques. Alternatively such systems, can also be described in frequency domain. Although not important for simulation purpose (sampling), frequency domain analysis offers many methods for system stability bandwidth and damping analysis.

Such aspects certainly provide also important medical information about a physiological system. E.g. if one would think of correlation of elasticity and damping of a calcified vessel or a long time convergence behavior of a critical vital signal after a medication.

The above mentioned aspects play a significant role in physiological system analysis, which is usually an off-line process, meaning that no real-time simulation conditions are given. Such analysis easily provided by Fourier Transformation was not in the focus of this work, but can easily be added to the current environment, as a feature for model analysis, e.g. during debriefing phases.

10.2.4 System Initialization

As proposed by our methods first order linear differential equations have been used to describe dynamic systems. The formalism of dynamic equations is unified in the time scale calculus theory [11]. For linear differential equations the initial conditions have to be known. For first order differential equations, this means the value of the system states at the beginning of the simulation for the time t_0. For an n-dimensional dynamic system $\frac{dx_i}{dt} = f_i(t, x_1, ..., x_n)$, $(t, x_1, ..., x_n) \in R_+ \times R^n, i = 1, .., n$, the stability is guaranteed for non degenerate systems, in other words if the following condition is fulfilled [40]:

$$det\left(\frac{\partial f_i(t, 0, ..., 0)}{\partial x_i}\right) \neq 0 \tag{10.1}$$

Essentially eigenvalues of the system matrix determine the structure of the phase space. Thus the eigenvalues and the eigenvectors determine if an initial point will converge or diverge.

The purpose of our system is not to analyze system stability of dynamic systems, because it will mostly be used for real-time simulations of physiological outcome in medical education. Therefore we expect the physiological models, described by XML, and loaded to the environment, to describe stable system properties.

Essentially for physiological compartment modeling, due to clearance from the compartments, usually the systems are stable per se. Addtionaly the modeler has to provide the initial value conditions, which is also by definition given, e.g. by the concentration of a medicament at the begin of a medication procedure.

Because of the combination of different modeling approaches, the resulting systems can be highly non-linear. This is already the case for non uniformly continuous functions, which can be specified in our models by "if-then relations". Also here system stability analysis for non-linear systems would go beyond the scope of this thesis.

However, due to the flexible design of the simulation software, one can easily embedded necessary system analysis components if necessary, to simplify model design processes.

10.2.5 System Synchronization

In fact physiological modeling can become a multi-resolution and multi-scale problem. Multi-resolution in time domain is e.g. given, because for some processes the transitions between

discrete states (even if they are assumed to be continuous, by realization on a digital computing system, they are implicitly discrete) have different timing granularities. For example some regulation processes are important after hours, others after milliseconds.

Some approaches like SAPHIR [223] spend a lot of efforts to formally describe spatial coupling and temporal synchronization. By this, a global synchronization with a fixed time step and local simulation with an adaptive time step was realized. Although this is certainly a good approach especially for distributed executions on computer clusters, models in SAPHIR are realized by a fixed realization in source-code and thus are very hard to modify for a non-programming user.

In our approach we encapsulate the modeling with XML from the computation and execution, by this the modeler has more freedom to also decide, if synchronization is necessary or not. If nothing is specified additionally a fixed global step size is used for all sub-models and thus synchronization is not necessary. However, the user can decide to run a model with an integer multiply of the global sample time.

This constrains have been specified, because we need to support a real-time capable simulation and according to the opinion of the author synchronization mechanisms have to been managed by the modeler to guarantee maximal flexibility.

10.2.6 Complexity

By the proposed methods, we have provided strategies and a feasible platform, to model complex physiological processes with different mathematical foundations in a hierarchical composition.

Herewith the complexity of the design and modeling processes is addressed in two ways. First, due to the hierarchical composition, huge models can be divided in smaller submodes, which lead to a reduction of the complexity from model to sub-model level. Second, due to novel combination and integration of static and dynamic BN the new environment provides a new and flexible way to describe physiological processes based on probabilistic influences and causal relations (BNs are probabilistic causal graphs). This offers easy integration of models, based on expert knowledge and real-data representation by BN learning, leading to a simplified and compact model representation.

However, one has not to mistake the difference of modeling complexity with computational complexity. The latter was not the main address of this work, although due to the real-

time requirements some optimization strategies have been applied in the system realization. Logically, the overall computational model complexity will no decrease using composition techniques.

The complexity of physiological models will grow, due to the integration of different levels and this optimization will be an important aspect, addressing special purpose architectures, parallel and distributed computing e.g. grid or GPU accelerated computing.

11 Summary And Future Work

11.1 Summary

Novel methods for simulation and emulation of physiological vital signals have been proposed. The strategies aims at substantial improvements of medical educational simulator systems. However, the proposed methods are useful in general for physiological modeling and simulations.

By the hierarchical object oriented description of subsystems a novel way to describe more complex physiological systems is delivered. By this way, interventions, medications and complications can be described as additional independent systems and thus can be loaded in the simulation on demand. These event based modeling and simulation opens new perspectives for monitoring and prognostic of physiological modifications, which have been analyzed during this work.

The architecture of the simulator system including detailed views on hardware, software and logical structures, necessary to model a human patient physiological system, is presented during the work. The formalism is kept in a generic way, that guarantees extensibility and adaptivity on demand. The feasibility of the approaches is demonstrated by providing physical hardware devices and software implementations in c++.

We have shown that static and dynamic BNs are very useful for modeling and simulation of medical systems. The concepts have been demonstrated on examples from cognitive and psychomotoric and physiological modeling. The applicability for medical educational simulators was shown by integration into a human patient simulator.

By the hybrid approach, which is suggested during this work, probabilistic methods are used to classify or sample states of dynamic systems, which are combined with deterministic modeling approaches of dynamic systems. Due to hierarchical composition, complex systems can be divided to sub-models, which can be driven by different modeling techniques. This novel combination, which is not provided by other systems yet, has a significant advantage. Espe-

cially for physiological modeling, were different levels of detail are usually considered (organ, tissue, molecular, ...), those sub-models, where deterministic models lead to poor results (e.g. because of missing knowledge), probabilistic causal networks can be used for a better system description. Additionally, probabilistic causality is delivering a smart way to deal with uncertainty in model parameters, data and structure.

11.2 Future Work

According to the fact, that the current work is acting like a snowball effect, by the meaning of providing a basis for much more physiological considerations and the topic is very actual in virtual physiological human[1] modeling, future work and perspectives are ad infinitum. Never the less in the following some future directions are discussed briefly.

11.2.1 Psychology

According to the fact that psychological simulations are becoming more and more important for practical problems, not only in clinical domains such as clinical patient monitoring or clinical simulations, but also for any form of system including the human factor, focusing of appropriate approaches to model cognitive systems and describe correlations to real data will be an very important and innovative direction for future research. Especially psychological effects to the physiological system and medical health states can be considered in more detail and is also not covered sufficiently. We have provided basic methods for interdisciplinary psychological research and modeling problems and hope to address new ideas in this field. DBN will provide a serious alternative to classical and currently established SEM approaches.

11.2.2 Integration To Other Simulators

The PhysioSim simulator presented in this work is a very good basis for integration of physiological modeling into other educational simulators. The concept of XML based model description is allowing other contributors to design appropriated physiological models according to the purposes of the particular educational simulator. Addtionaly, using client and server communication the PhysioSim server functionality can run independently on a distributed

[1] See also the virtual physiological human network of excellence as part of the EU research call 7 http://www.vph-noe.eu/

computing system and provide the necessary signals for integration into a standalone simulator.

To underline the compatibility of the PhysioSim platform, integration to other simulator systems is planned in near future. On the one hand, cardiovascular physiological models, e.g. given by haemodynamics are very useful for the cardiovascular intervention simulator Cathi [37]. Here an integration and extension of current capabilities is planed in 2010. Additionally we are cooperating with the St George's University, London, to establish a high fidelity' virtual human patient simulator of the cardiovascular and respiratory systems, based on the PhysioSim methodologies.

An important part of this work was also focusing on physiological hardware emulation interfaces for vital signals and corresponding models and strategies. We have shown, that this devices can be realized and can compete with existing hardware of other part- and full-scale simulators. Although virtual patients and virtual patient physiology is becoming more and more important, dummy patient simulators and their corresponding hardware are still important for efficient learning of medical skills. We hope that the provided hardware will deliver additional realism in learning to deal with vital signal monitoring and understanding skills.

11.2.3 Prediction Of Patient States

As it has been shown in chapter 9, physiological simulations can be used for much more than educational purpose. Nowadays off-line simulations assists in clinical decision making as well as for treatment planning. On-line simulations can lead to a new generation of vital monitoring systems. One important aspect is the ability to provide predictions for future outcome of patient states as discussed in chapter 7.

To analyze the practical abilities of PhyioSim software environment, a project has already started with Neuroradiological department of the university clinic in Mannheim to model cerebral haemodynamics and provide a physiological based monitoring of the vasospasm process for clinical usage, based on real-time simulation.. By this approach, physicians should be able to provide better predictions for future vasospasm state and provide appropriate medications strategies.

As it is also the intension of other groups, e.g. the Physiome project, physiological modeling and simulation should be more connected to functional imaging methods, by integrating the knowledge about the physiology.

11.2.4 Optimization

Due to the increasing complexity of physiological models as well as the amount of patient data, optimization of computational processes is a very important aspect. This is as well valid for for deterministic solvers as for probabilistic solvers integrated into this project.

In this project GSL [87] ODE solver methods have been used to solve the system of linear differential equations. However new methods show that potential improvements, especially for big systems of ODEs can be achieved [111], e.g. linear time complexity instead of polynomial one.

Due to the space and time complexity for multiple DBN slices, the current DBN are limited in number of nodes. Therefore various groups focus on optimization techniques reducing the size of the network kept in memory to perform calculations. Most of the methods apply approximative techniques, e.g. by removing weak dependencies between different time slices. New methods using parallel computing techniques, show significant acceleration in inference techniques.

Todays efforts to integrate BN analysis into embedded systems is accelerating the research in this area, because of the less computational power and memory available on embedded system platforms.

Due to the fact that our models are becoming increasingly huge for full scale physiological simulations one practical topic for future work will be the optimization of inference algorithms of DBN for real-time systems, although already the hierarchical modeling approach can be used to run the sub models on different CPU cores.

11.2.5 Standards

The huge amount of different modeling tools existing around the world, underlines the importance of a way to exchange models, independent from the simulation tools. Among others SBML [202] is the most promising way to define a standard for physiological modeling. Although also this language is facing restrictive problems, e.g. because it cannot be general in high level modeling and the same time focus on low level models, it seems to be very adaptive to new methods and is also planning to provide hierarchical modeling.

Nevertheless the integration of the particular models, e.g. for kidney, circulation, metabolism into a full-, part-scale or virtual simulator based on a standardized language and way is

unfortunately still far from todays reality. But exactly this step will lead to significant improvements of current simulator standards around the world. One has to imagine that at the moment each simulator company and each simulator center is using its own non-standard physical models, although they are usually based on a unique logical idea, e.g. the circulatory modeling of Guyton [88]. Especially integrative models - combining different models to one huge and complex - are very design specific and also very rare. Successful realizations like the SAPHIR [223] or QCP [186] project unfortunately are again not compatible to SBML yet and thus difficult to export. SBML itself maybe needs to also consider extensions toward other modeling mechanisms, like probabilistic causal networks to cover more modeling freedom.

This process, which we are planning to participate in, within the next European Virtual Human Physiology collaboration, will also affect the problems with integration of non-standardized ways of PBPKPD models into simulation systems.

11.2.6 Interface Applications

Huge amount of uncorrelated medical data are existing. Old linear regression methods are unfortunately still typical in clinical praxis, however they loose lots of information. Automatic processes to find important correlations are not given, despite Gaussian and regression analysis tools. Bayesian networks provide here, a significant advantages, because they support the intervention of the modeler during the model development on the one hand and provide multivariate approximation of probability density functions on the other hand, providing a better model fit. Therefore probabilistic causality is gaining attention in medical daily use in the last years.

Nevertheless, automatic BN learning from real data, tailored to the clinical and medical demands, is till a desired feature. In the 9th ACSI conference of software engineering and artificial intelligence in 2008 a first approach for dynamic data feed was presented, which could play an magnificent role for future research in this direction [123].

Correlation of image acquisition with physiological parameters based on physiological simulations will be the focus in the near future research. This topic of functional imaging is becoming more and more important during the last years. Also DBNs seems to be applicable for complex relationship modelings as done for cerebral activity network (functional neuroimaging) based on PET image acquisition [132].

Dependability analysis is an important topic, which is unfortunately less examined in clinical and medical domain yet, although significant strategies have been presented in the last

years. The hybrid method, presented in this work, to describe dynamic systems as well as dependability analysis using DBN are per se interesting aspects for future research. Hopefully this topic and the delivered strategies will gain more attention to increase patient safety in medicine during the next years of research.

Acronyms

ACRM	Anaesthetic Crisis Resource Management
AI	Artificial Intelligence
ANN	Artificial Neural Network
ANS	Autonomic Nervous System
BN	Bayesian Network
CO	Cardiac Output
CPD	Conditional Probability Distribution
CPT	Conditional Probability Table
CTBN	continuous time Bayesian network
CVS	Cardio Vascular System
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network
DNA	Deoxyribo Nucleic Acid
DSL	Decision Systems Laboratory
DSS	Decision Support System
ECG	Electrocardiogram
EM	Expectation Maximization
GUI	Graphical User Interface
HANS	"Heidelberg Anaesthesie- und Notfall-Simulator"
HMI	Human Machine Interaction
HMM	Hidden Markov Model
HPS	Human Patient Simulator
HR	Heart Rate
IBP	Invasive Blood Pressure

ICT	Information and Communication Technology
IDE	Integrated Development Environment
IUPS	International Union of Physiological Sciences
JPD	Joint Probability Distribution
LED	Light Emitting Diode
LOS	Level of Significance
LPD	Local Probability Distribution
MathML	Mathematical Markup Language
MML	Mathematical Modeling Language
MVC	Model View Control
MVP	Maryland Virtual Patient
NLP	Natural Language Processing
ODE	Ordinary Differential Equation
PBN	Prognostic Bayesian Network
PBPD	Physiologically Based Pharmaco-Dynamics
PBPK	Physiologically Based Pharmaco-Kinetics
PD	Pharmacodynamic
PDF	Probability Density Function
PK	Pharmacokinetic
QCP	Quantitative Circulatory Physiology
QMR	Quick Medical Reference
QoS	Quality of Service
SaO2	Saturation of Oxygen
SBML	Systems Biology Markup Language
SEM	Structural Equation Modeling
VPH	Virtual Physiological Human
VR	Virtual Reality
XML	Extensible Markup Language

List of Figures

List of Tables

Bibliography

[1] Rescigno A. The rise and fall of compartmental analysis. *Pharmacological Research*, 44:337–342, October 2001.

[2] L Aarons. Population pharmacokinetics: theory and practice. *British Journal of Clinical Pharmacology*, 32(6):669–670, December 1991. PMID: 1768557.

[3] L. Aarons. Physiologically based pharmacokinetic modelling: a sound mechanistic basis is needed. *British Journal of Clinical Pharmacology*, 60(6):581–583, December 2005. PMC1884881.

[4] Ciamak Abkai, Nico Becherer, Jürgen Hesser, and Reinhard Manner. Real-time simulator for intravascular ultrasound (IVUS). In *Proceedings of SPIE*, pages 65131E–65131E–11, San Diego, CA, USA, 2007.

[5] Ciamak Abkai and Jürgen Hesser. Real-time ECG emulation: a multiple dipole model for electrocardiography simulation. *Studies in Health Technology and Informatics*, 142:7–9, 2009. PMID: 19377101.

[6] Ciamak Abkai and Jürgen Hesser. Virtual intensive care unit (ICU): real-time simulation environment applying hybrid approach using dynamic bayesian networks and ODEs. *Studies in Health Technology and Informatics*, 142:1–6, 2009. PMID: 19377100.

[7] Ciamak Abkai and Juergen Hesser. Real-time physiological simulation and modeling toward dependable patient monitoring systems. In *International Workshop on the Design of Dependable Critical Systems, HeiDok : 978-3-00-029877-6*, 2009.

[8] Ciamak Abkai and Juergen Hesser. Reasoning and risk assessment based on Real-Time physiological simulations and dependable vital parameter monitoring. In *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany*, pages 1299–1302. Springer Berlin Heidelberg, 2009.

[9] Sean R. Abram, Benjamin L. Hodnett, Richard L. Summers, Thomas G. Coleman, and

Robert L. Hester. Quantitative circulatory physiology: an integrative mathematical model of human physiology for medical education. *Advan. Physiol. Edu.*, 31(2):202–210, June 2007.

[10] Joel C. Adams. OOP and the janus principle. *SIGCSE Bull.*, 38(1):359–363, 2006.

[11] Ravi Agarwal, Martin Bohner, Donal O'Regan, and Allan Peterson. Dynamic equations on time scales: a survey. *J. Comput. Appl. Math.*, 141(1-2):1–26, 2002.

[12] J Allard, S Cotin, F Faure, P-J Bensoussan, F Poyer, C Duriez, H Delingette, and L Grisoni. SOFA–an open source framework for medical simulation. *Studies in Health Technology and Informatics*, 125:13–18, 2007. PMID: 17377224.

[13] Paul Root Wo Amitai Ziv, Stephen D. Small. Patient safety and simulation-based medical education. *Medical Teacher*, 22(5):489–495, 2000.

[14] J N Amoore, W P Santamore, W J Corin, and D T George. Computer simulation of the effects of ventricular interdependence on indices of left ventricular systolic function. *Journal of Biomedical Engineering*, 14(3):257–262, May 1992. PMID: 1588784.

[15] Anesoft. Anesthesia and critical care software: http://www.anesoft.com/.

[16] J. Mark Ansermino, Jeremy P. Daniels, Randy T. Hewgill, Joanne Lim, Ping Yang, Chris J. Brouse, Guy A. Dumont, and John B. Bowering. An evaluation of a novel software tool for detecting changes in physiological monitoring. *Anesth Analg*, 108(3):873–880, March 2009.

[17] Juan Carlos Augusto. Temporal reasoning for decision support in medicine. *Artificial Intelligence in Medicine*, 33(1):1–24, 2005.

[18] Algirdas Aviaienis, Jean-Claude Laprie, and Brian Randell. Dependability and its threats: A taxonomy. In *Building the Information Society*, pages 91–120. Kluwer, 2004.

[19] N. Ayache and D. Chapelle. Towards Model-Based estimation of the cardiac Electro-Mechanical activity from ECG signals and ultrasound images. In *Functional Imaging and Modeling of the Heart*, pages 120–127. 2001.

[20] Cheng-Gang Bai. Bayesian network based software reliability prediction with an operational profile. *J. Syst. Softw.*, 77(2):103–112, 2005.

[21] S J Barker and K K Tremper. Pulse oximetry: applications and limitations. *International Anesthesiology Clinics*, 25(3):155–175, 1987. PMID: 3323062.

[22] Nico Becherer. Simulation von komplikationen bei herzkathetereingriffen, diploma thesis, university of mannheim, germany, 2004.

[23] IA Beinlich, HJ Suermondt, RM Chavez, and GF Cooper. The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks. In *Second European Conference on Artificial Intelligence in Medicine*, volume 38, pages 256, 247. Springer-Verlag, Berlin, 1989.

[24] Sven Bjorkman. Prediction of drug disposition in infants and children by means of physiologically based pharmacokinetic (PBPK) modelling: theophylline and midazolam as model drugs. *British Journal of Clinical Pharmacology*, 59(6):691–704, 2005.

[25] Phil Blyth. *Virtual Reality Simulation of Hip Surgery*. PhD thesis, University of Auckland, 2008.

[26] BodySim. Body simulation : http://www.advsim.com/.

[27] H. Boudali and J.B. Dugan. A discrete-time bayesian network reliability modeling and analysis framework. *Reliability Engineering & System Safety*, 87(3):337–349, March 2005.

[28] F. Douglas Boudinot, Robin D'Ambrosio, and William J. Jusko. Receptor-mediated pharmacodynamics of prednisolone in the rat. *Journal of Pharmacokinetics and Pharmacodynamics*, 14(5):469–493, October 1986.

[29] Xavier Boyen and Daphne Koller. Tractable inference for complex stochastic processes. *IN PROC. UAI*, pages 33–42, 1998.

[30] BRAINCIRC. Modelling the human brain circulation, http://braincirc.sourceforge.net/.

[31] Benny Bürger, Ciamak Abkai, and Jürgen Hesser. Simulation of dynamic ultrasound based on CT models for medical education. *Studies in Health Technology and Informatics*, 132:56–61, 2008. PMID: 18391257.

[32] C. Brouse, G. Dumont, Ping Yang, J. Lim, and J.M. Ansermino. iAssist: a software framework for intelligent patient monitoring. In *Engineering in Medicine and Biology*

Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE, pages 3790–3793, 2007.

[33] Eugene N. Bruce and Margaret C. Bruce. A multicompartment model of carboxyhemoglobin and carboxymyoglobin responses to inhalation of carbon monoxide. *J Appl Physiol*, 95(3):1235–1247, September 2003.

[34] E A Byrne and R Parasuraman. Psychophysiology and adaptive automation. *Biological Psychology*, 42(3):249–268, February 1996. PMID: 8652747.

[35] Yang Cai. *Digital Human Modeling*. Springer, 2008.

[36] Brian E Carlson, Joseph C Anderson, Gary M Raymond, Ranjan K Dash, and James B Bassingthwaighte. Modeling oxygen and carbon dioxide transport and exchange using a closed loop circulatory system. *Advances in Experimental Medicine and Biology*, 614:353–360, 2008. PMID: 18290346.

[37] CATHI. Catheter instruction system: http://www.cathi.eu/.

[38] CellML. http://www.cellml.org/.

[39] Vincent Chan and Steve Underwood. A Single-Chip pulsoximeter design using the MSP430 application report : Technical documents from texas instruments. Technical report, Texas Instruments, 2005.

[40] Zhu Chang-jiang. Initial value problem for high dimensional dynamic systems. *Applied Mathematics and Mechanics*, 16(3):279–282, March 1995.

[41] Nicolas W. Chbat and Kees Van Zon. US patent 20090254328, dynamic bayesian network for emulating cardiovascular function, October 2009.

[42] David C. Chung. *Ventricular interaction in a closed-loop model of the canine circulation*. Master thesis, Rice University, 1996.

[43] Barry R. Cobb and Prakash P. Shenoy. Operations for inference in continuous bayesian networks with linear deterministic variables. *International Journal of Approximate Reasoning*, 42(1-2):21–36, May 2006.

[44] E Coiera. Intelligent monitoring and control of dynamic physiological systems. *Artificial Intelligence in Medicine*, 5(1):1–8, February 1993. PMID: 8358484.

[45] T G Coleman and J E Randall. HUMAN. a comprehensive physiological model. *The Physiologist*, 26(1):15–21, February 1983. PMID: 6601809.

[46] TG Coleman and JE Randall. A role for mathematical models and computer simulation in the teaching of physiology. *The Physiologist*, 26(4):231, 1983.

[47] STEP Consortium. Seeding the europhysiome: A roadmap to the virtual physiological human, http://www.europhysiome.org/roadmap, 2007.

[48] Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks (research note). *Artif. Intell.*, 42(2-3):393–405, 1990.

[49] J B Cooper and V R Taqueti. A brief history of the development of mannequin simulators for clinical education and training. *Postgraduate Medical Journal*, 84(997):563–570, November 2008.

[50] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms, Second Edition*. MIT Press, 2 edition, September 2001.

[51] Stephane Cotin, David W Shaffer, Dwight A Meglan, Mark P Ottensmeyer, Patrick S Berry, and Steven L Dawson. CAML: a general framework for the development of medical simulation systems. http://adsabs.harvard.edu/abs/2000SPIE.4037..294C, August 2000.

[52] Edmund J Crampin, Matthew Halstead, Peter Hunter, Poul Nielsen, Denis Noble, Nicolas Smith, and Merryn Tawhai. Computational physiology and the physiome project. *Experimental Physiology*, 89(1):1–26, 2004. PMID: 15109205.

[53] Meindert Danhof, Elizabeth C.M. de Lange, Oscar E. Della Pasqua, Bart A. Ploeger, and Rob A. Voskuyl. Mechanism-based pharmacokinetic-pharmacodynamic (PK-PD) modeling in translational drug research. *Trends in Pharmacological Sciences*, 29(4):186–191, April 2008.

[54] Denver Dash. *Caveats for Causal Reasoning with Equilibrium Models*. PhD thesis, University of Pittsburgh, May 2003.

[55] Cassio P. de Campos, Zhi Zeng, and Qiang Ji. Structure learning of bayesian networks using constraints. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 113–120, Montreal, Quebec, Canada, 2009. ACM.

[56] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Comput. Intell.*, 5(3):142–150, 1990.

[57] H. Delingette, X. Pennec, L. Soler, J. Marescaux, and N. Ayache. Computational models for Image-Guided Robot-Assisted and simulated medical interventions. *Proceedings of the IEEE*, 94(9):1678–1688, 2006.

[58] AP Dempster, NM Laird, and DB Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):38, 1, 1977.

[59] J. S. Denson and Stephen Abrahamson. A Computer-Controlled patient simulator. *JAMA*, 208(3):504–508, April 1969.

[60] Rainer Deventer, Joachim Denzler, and Heinrich Niemann. Control of dynamic systems using bayesian networks. *Proceedings of the IBERAMIA/SBIA 2000 Workshops (Atibaia, Sdo Paulo*, 2000.

[61] Amira Djebbari and John Quackenbush. Seeded bayesian networks: constructing genetic networks from microarray data. *BMC systems biology*, 2(1):57, 2008.

[62] Johnson Don, PhD, Flagg Amanda, and Dremsa Theresa L. Effects of using human patient simulator (HPS) versus a CD-ROM on cognition and critical thinking. *Medical Education Online*, 13, 2008.

[63] M. Dosani, J. Lim, P. Yang, C. Brouse, J. Daniels, G. Dumont, and J. M. Ansermino. Clinical evaluation of algorithms for context-sensitive physiological monitoring in children. *Br. J. Anaesth.*, page aep045, March 2009.

[64] JM Downey and ES Kirk. Inhibition of coronary blood flow by a vascular waterfall mechanism. *Circ Res*, 36(6):753–760, June 1975.

[65] Doxygen. Documentation generation software, http://www.doxygen.org/.

[66] DSL. Decision systems laboratory, http://dsl.sis.pitt.edu/.

[67] Paul M. Duvall, Steve Matyas, and Andrew Glover. *Continuous Integration: Improving Software Quality and Reducing Risk*. Addison-Wesley, July 2007.

[68] H A el-Masri, R S Thomas, S A Benjamin, and R S Yang. Physiologically based phar-

macokinetic/pharmacodynamic modeling of chemical mixtures and possible applications in risk assessment. *Toxicology*, 105(2-3):275–282, December 1995. PMID: 8571364.

[69] EuroPhysiome. http://www.europhysiome.org/.

[70] HUGIN EXPERT. Provider of software for advanced decision support based on complex statistical models, bayesian networks (bn), http://www.hugin.com/.

[71] Michael Fisher, Dov M. Gabbay, and LluÃs Vila. *Handbook of temporal reasoning in artificial intelligence*. Elsevier, 2005.

[72] Nir Friedman. The Bayesian Structural EM Algorithm. In *UAI*, 1998.

[73] Nir Friedman, Moisas Goldszmidt, and Thomas J. Lee. Bayesian network classification with continuous attributes: Getting the best of both discretization and parametric fitting. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 179–187. Morgan Kaufmann Publishers Inc., 1998.

[74] Nir Friedman and Moises Goldszmidt. Discretizing continuous attributes while learning bayesian networks. *IN PROC. ICML*, pages 157–165, 1996.

[75] Frost and Sullivan. Return on investment study for medical simulation training. Technical report, Immersion Medical, Inc. Laparoscopy AccuTouch System., 2004.

[76] David M. Gaba, Steven K. Howard, Kevin J. Fish, Brian E. Smith, and Yasser A. Sowb. Simulation-based training in anesthesia crisis resource management (ACRM): a decade of experience. *Simul. Gaming*, 32(2):175–193, 2001.

[77] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1. edition, 1995.

[78] GasMan. Anesthesia simulation, http://www.gasmanweb.com/.

[79] M.C.F. Geven and V.N. Bohte. A physiologically representative in vitro model of the coronary circulation. *Physiological Measurement*, 25(4):891–904, August 2004. PMID: 15382829.

[80] Jayanta K. Ghosh and Marco Valtorta. Building a bayesian network model of heart disease. In *Proceedings of the 38th annual on Southeast regional conference*, pages 239–240, Clemson, South Carolina, 2000. ACM.

[81] Bayer Technology Services GmbH. Systems biology : Pk-sim : The comprehensive software tool for whole-body physiologically based pharmacokinetic (wb-pbpk) modeling and simulation., http://www.systems-biology.com.

[82] Sofie Goderis. DEUCE: a declarative framework for extricating user interface concerns. *Journal of Object Technology, Special Issue: TOOLS*, pages 87–104, 2007.

[83] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13). Circulation Electronic Pages: http://circ.ahajournals.org/cgi/content/full/101/23/e215.

[84] M L Good. Patient simulation for training basic and advanced clinical skills. *Medical Education*, 37 Suppl 1:14–21, November 2003. PMID: 14641634.

[85] O. Grottke, A. Ntouba, S. Ullrich, W. Liao, E. Fried, A. Prescher, T. M. Deserno, T. Kuhlen, and R. Rossaint. Virtual reality-based simulator for training in regional anaesthesia. *Br. J. Anaesth.*, 103(4):594–600, October 2009.

[86] C. Grube, S. Volk, Y. Zausig, and B.M. Graf. Changing culture -simulatortraining als mittel zur erhöhten patientensicherheit bericht vom international meeting on medical simulation, scottsdale, 12.-14.1.2001. *Der Anaesthesist*, 50(5):358–362, May 2001.

[87] GSL. GNU scientific library, www.gnu.org/software/gsl/.

[88] A C Guyton, T G Coleman, and H J Granger. Circulation: overall regulation. *Annual Review of Physiology*, 34:13–46, 1972. PMID: 4334846.

[89] A C GUYTON, A W LINDSEY, and B N KAUFMANN. Effect of mean circulatory filling pressure and other peripheral circulatory factors on cardiac output. *The American Journal of Physiology*, 180(3):463–468, March 1955. PMID: 14376522.

[90] AC Guyton, CE Jones, and TG Coleman. CIRCULATORY PHYSIOLOGY: cardiac output and its regulation. Text, Philadelphia, 1963.

[91] Arthur C. Guyton and John E. Hall. *Textbook of Medical Physiology*. Saunders W.B., 11. a. edition, July 2005.

[92] Ernst Hairer, Syvert P. NÃ¸rsett, and Gerhard Wanner. *Solving Ordinary Differential*

Equations I: Nonstiff Problems. Springer, Berlin, 2nd ed. 1993. corr. 3rd printing. 1993. corr. 3rd. edition, December 2009.

[93] Brigette M. Hales and Peter J. Pronovost. The checklist–a tool for error management and performance improvement. *Journal of Critical Care*, 21(3):231–235, September 2006.

[94] John E. Hall. The pioneering use of systems analysis to study cardiac output regulation. *Am J Physiol Regul Integr Comp Physiol*, 287(5):R1009–1011, November 2004.

[95] Haptica. http://www.haptica.com/.

[96] J. G. Hardman, N. M. Bedforth, A. B. Ahmed, R. P. Mahajan, and A. R. Aitkenhead. A physiology simulator: validation of its respiratory components and its ability to predict the patient's response to changes in mechanical ventilation. *Br. J. Anaesth.*, 81(3):327–332, September 1998.

[97] H. Hardy and R. Collins. On the pressure-volume relationship in circulatory elements. *Medical and Biological Engineering and Computing*, 20(5):565–570, 1982.

[98] SECOND HEALTH. A virtual health center by national health service london, http://secondhealth.wordpress.com/.

[99] David Heckerman, Dan Geiger, and David M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.*, 20(3):197–243, 1995.

[100] University Heidelberg. University clinic, http://www.klinikum.uni-heidelberg.de/.

[101] Blanca E. Himes, Yi Dai, Isaac S. Kohane, Scott T. Weiss, and Marco F. Ramoni. Prediction of chronic obstructive pulmonary disease (COPD) in asthma patients using electronic medical records. *J Am Med Inform Assoc*, 16(3):371–379, May 2009.

[102] P H Hinderling, U Gundert-Remy, and O Schmidlin. Integrated pharmacokinetics and pharmacodynamics of atropine in healthy humans. i: Pharmacokinetics. *Journal of Pharmaceutical Sciences*, 74(7):703–710, July 1985. PMID: 4032240.

[103] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, and the rest of the SBML forum. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, March 2003.

[104] Michael Hucka and Andrew Finney. Escalating model sizes and complexities call for

standardized forms of representation. *Molecular Systems Biology*, 1:2005.0011, 2005. PMID: 16729046.

[105] Hudson. An extensible continuous integration software, https://hudson.dev.java.net/.

[106] Joris Hulst. Modeling physiological processes with dynamic bayesian networks, September 2006.

[107] J. D. Humphrey, S. Baek, and L. E. Niklason. Biochemomechanics of cerebral vasospasm and its resolution: I. a new hypothesis and theoretical framework. *Annals of Biomedical Engineering*, 35(9):1485–1497, 2007.

[108] Peter J. Hunter and Thomas K. Borg. Integration from proteins to organs: the physiome project. *Nat Rev Mol Cell Biol*, 4(3):237–243, March 2003.

[109] Andrew Hussey, Andrew Hussey, David Carrington, and David Carrington. Comparing two user-interface architectures: MVC and PAC. *MVC AND PAC. FAHCI'96*, pages 3–21, 1996.

[110] Trey Ideker, Timothy Galitski, and Leroy Hood. A NEW APPROACH TO DECODING LIFE: systems biology. *Annual Review of Genomics and Human Genetics*, 2(1):343–372, 2001.

[111] Silvana Ilie, Robert M. Corless, and Greg Reid. Numerical solutions of index-1 differential algebraic equations can be computed in polynomial time. *Numerical Algorithms*, 41(2):161–171, 2005.

[112] Texas Instruments. Msp430x1xx family - user guide - slau049f, http://focus.ti.com/. Technical report, TI, 2006.

[113] Suminobu Ito, Takeshi Kawamura, Makoto Inada, Yoshiharu Inoue, Yukihiro Hirao, Toshihisa Koga, Jun ichi Kunizaki, Takefumi Shimizu, and Hitoshi Sato. Physiologically based pharmacokinetic modelling of the three-step metabolism of pyrimidine using c-uracil as an in vivo probe. *British Journal of Clinical Pharmacology*, 60(6):584–593, December 2005. PMID: 16305582.

[114] IUPS. http://www.physiome.org.nz/.

[115] Ari Jaaksi and Ari Jaaksi. Implementing interactive applications in c. 1995.

[116] Adolf Jaeger, Heinz-Martin Suess, and Andre Beauducel. *Berliner Intelligenzstruktur-Test (BIS-Test).* Hogrefe Verlag, 1997.

[117] Bruce Jarrell, Sergei Nirenburg, Marjorie McShane, George Fantry, Stephen Beale, David Mallott, and John Raczek. An interactive, cognitive simulation of gastroesophageal reflux disease. *Studies in Health Technology and Informatics*, 125:194–199, 2007. PMID: 17377265.

[118] Finn V. Jensen. *Bayesian Networks and Decision Graphs.* Springer, corrected edition, May 2002.

[119] Mingfeng Jiang, Ling Xia, and Guofa Shou. Combining regularization frameworks for solving the electrocardiography inverse problem. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*, pages 1210–1219. 2007.

[120] M. Jipp, E. Badreddin, C. Abkai, and J. Hesser. Individual ability-based system configuration cognitive profiling with bayesian networks. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 3359–3364, 2008.

[121] Meike Jipp, Ciamak Abkai, and et al. Psychomotor profiling with bayesian neworks: Prediction of user abilities based on inputs of motorized wheelchair parameters. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 2009.

[122] Doyle D. John. Simulation in medical education: Focus on anesthesiology and critical care medicine. http://eprints.kfupm.edu.sa/127664/, 2002.

[123] Nipat Jongsawat, Pittaya Poompuang, and Wichian Premchaiswadi. Dynamic data feed to bayesian network model and SMILE web application. In *Proceedings of the 2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pages 931–936. IEEE Computer Society, 2008.

[124] Markus Kalisch and Peter Bühlmann. Estimating High-Dimensional directed acyclic graphs with the PC-Algorithm. *J. Mach. Learn. Res.*, 8:613–636, 2007.

[125] James P. Keener and James Sneyd. *Mathematical Physiology.* Springer, October 2008.

[126] Uffe Kjaerulff. Reduction of computational complexity in bayesian networks through

removal of weak dependences. *IN PROC. TENTH CONF. ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE*, pages 374–382, 1994.

[127] Linda T. Kohn, Janet Corrigan, Molla S. Donaldson, and Institute of Medicine (U.S.). Committee on Quality of Health Care in America. *To err is human*. National Academies Press, 2000.

[128] Kevin B. Korb, Ann Nicholson, and Korb B. Korb. *Bayesian Artificial Intelligence*. Chapman & Hall/CRC, illustrated edition edition, September 2003.

[129] Glenn E. Krasner and Stephen T. Pope. A cookbook for using the model-view controller user interface paradigm in smalltalk-80. *J. Object Oriented Program.*, 1(3):26–49, 1988.

[130] David N. Ku. BLOOD FLOW IN ARTERIES. *Annual Review of Fluid Mechanics*, 29(1):399–434, 1997.

[131] Kevin Kunkler. The role of medical simulation: an overview. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 2(3):203–210, 2006.

[132] Vincent Labatut, Josette Pastor, Serge Ruff, Jean-FranÃ§ois DÃ©monet, and Pierre Celsis. Cerebral modeling and dynamic bayesian networks. *Artificial Intelligence in Medicine*, 30(2):119–139, February 2004.

[133] Laerdal. http://www.laerdal.com.

[134] Samsun (Sem) Lampotang. Computer and web-enabled simulations for anesthesiology training and credentialing. *Journal of Critical Care*, 23(2):173–178, June 2008.

[135] J. Lindsey Lane, Stuart Slavin, and Amitai Ziv. Simulation in medical education: A review. *Simulation Gaming*, 32(3):297–314, September 2001.

[136] Helge Langseth and Luigi Portinale. Bayesian networks in reliability. *Reliability Engineering & System Safety*, 92(1):92–108, 2007.

[137] Craig Larman. *Applying UML and Patterns*. Prentice Hall PTR, 3. edition, 2002.

[138] Steffen L Lauritzen and Steffen L Lauritzen. Propagation of probabilities, means and variances in mixed graphical association models. *JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION*, 87:1098–1108, 1992.

[139] M.S. Leaning, H.E. Pullen, E.R. Carson, and L. Finkelstein. Modelling a complex biolog-

ical system: the human cardiovascular system–1. methodology and model description. *Transactions of the Institute of Measurement and Control*, 5(2):71–86, April 1983.

[140] Hyun Jin Lee and W. E. Schiesser. *Ordinary and partial differential equation routines in C, C++, Fortran, Java, Maple, and MATLAB*. CRC Press, November 2003.

[141] J B Lemaire, J P Schaefer, L A Martin, P Faris, M D Ainslie, and R D Hull. Effectiveness of the quick medical reference as a diagnostic tool. *CMAJ: Canadian Medical Association Journal = Journal De l'Association Medicale Canadienne*, 161(6):725–728, September 1999. PMID: 10513280.

[142] Uri Lerner, Ronald Parr, Daphne Koller, and Gautam Biswas. Bayesian fault detection and diagnosis in dynamic systems. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 531–537. AAAI Press / The MIT Press, 2000.

[143] Geoffrey K. Lighthall and Juliana Barr. The use of clinical simulation systems to train critical care physicians. *J Intensive Care Med*, 22(5):257–269, September 2007.

[144] A. [1] Lindsay, K. [2] Lindsay, and J. [3] Rosenberg. New concepts in compartmental modelling. *Computing and Visualization in Science*, 10:79–98, June 2007.

[145] Mehmet Kayaalp Lister. Bayesian methods for diagnosing physiological conditions of human subjects from multivariate time series biosensor data. *IN PROCEEDINGS OF THE ICML PHYSIOLOGICAL DATA MODELING CONTEST WORKSHOP*, 2004.

[146] Catherine M. Lloyd, Matt D. B. Halstead, and Poul F. Nielsen. CellML: its future, present and past. *Progress in Biophysics and Molecular Biology*, 85(2-3):433–450, June.

[147] Carlo Alberto Lodi and Mauro Ursino. Hemodynamic effect of cerebral vasospasm in humans: A modeling study. *Annals of Biomedical Engineering*, 27(2):257–273, March 1999.

[148] Peter Lucas. Bayesian networks in medicine: a model-based approach to medical decision making. *Proceedings of the EUNITE workshop on Intelligent Systems in patient Care*, pages 73–97, October 2001.

[149] Peter J F Lucas, Linda C van der Gaag, and Ameen Abu-Hanna. Bayesian networks in biomedicine and health-care. *Artificial Intelligence in Medicine*, 30(3):201–14, March 2004. PMID: 15081072.

[150] Lawrence A. Lynn. US patent 6760608 - oximetry system for detecting ventilation instability.

[151] Sam Maes and Philippe Leray. Multi-agent causal models for dependability analysis. http://eprints.pascal-network.org/archive/00002104/, April 2006.

[152] I. Maglogiannis, E. Zafiropoulos, A. Platis, and C. Lambrinoudakis. Risk analysis of a patient monitoring system using bayesian network modeling. *J. of Biomedical Informatics*, 39(6):637–647, 2006.

[153] David Mallott, John Raczek, Craig Skinner, Kevin Jarrell, Mark Shimko, and Bruce Jarrell. A basis for electronic cognitive simulation: The heuristic patient. *Surgical Innovation*, 12(1):43–49, March 2005.

[154] Toru Masuzawa, Yasuhiro Fukui, and N. T. Smith. Cardiovascular simulation using a multiple modeling method on a digital computer Simulation of interaction between the cardiovascular system and angiotensin II. *Journal of Clinical Monitoring and Computing*, 8(1):50–58, March 1991.

[155] M. McShane, S. Nirenburg, and B. Jarrell. Maryland virtual patient: A Knowledge-Based, Language-Enabled simulation and training system - BAMS. In *Bio-Algorithms and Med-Systems*, volume 5 of *9*, pages 57–63, 2009.

[156] Marjorie Mcshane, Sergei Nirenburg, Stephen Beale, Bruce Jarrell, and George Fantry. Knowledge-Based modeling and simulation of diseases with highly differentiated clinical manifestations. In *Proceedings of the 11th conference on Artificial Intelligence in Medicine*, pages 34–43, Amsterdam, The Netherlands, 2007. Springer-Verlag.

[157] Patrick E McSharry, Gari D Clifford, Lionel Tarassenko, and Leonard A Smith. A dynamical model for generating synthetic electrocardiogram signals. *IEEE Transactions on Bio-Medical Engineering*, 50(3):289–294, March 2003. PMID: 12669985.

[158] Mentice. http://www.mentice.com.

[159] METI. http://www.meti.com/.

[160] Scott Meyers. *More Effective C++: 35 New Ways to Improve Your Programs and Designs*. Addison-Wesley Longman, Amsterdam, March 1996.

[161] V. Mihajlovic and M. Petkovic. Dynamic bayesian networks: A state of the art. http://doc.utwente.nl/36632/, 2001.

[162] Blake Madden Mike Dickheiser. *"Using CppUnit To Implement Unit Testing" in Game Programming Gems 6 (Book & CD-ROM)*. Charles River Media, 1 edition, March 2006.

[163] S. Mönk, H.-J. Baldering, J. Vollmer, H. Buggenhagen, and W. Heinrichs. Patientensimulation. *Notfall & Rettungsmedizin*, 2(5):297–306, July 1999.

[164] Jean-Pierre Montani and Bruce N. Van Vliet. Understanding the contribution of guyton's large circulatory model to long-term control of arterial pressure. *Experimental Physiology*, 94(4):382–388, April 2009.

[165] Kevin Murphy. Kevin murphy's website: http://people.cs.ubc.ca/ murphyk/.

[166] Kevin Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[167] Kevin P. Murphy and Yair Weiss. The factored frontier algorithm for approximate inference in DBNs. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 378–385. Morgan Kaufmann Publishers Inc., 2001.

[168] Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. *IN PROCEEDINGS OF UNCERTAINTY IN AI*, pages 467–475, 1999.

[169] M. Neil, M. Tailor, D. Marque, N. Fenton, and P. Hearty. Modeling dependable systems using hybrid bayesian networks. In *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*, page 5 pp., 2006.

[170] Sergei Nirenburg, Marjorie McShane, Stephen Beale, Bruce Jarrell, and George Fantry. Integrating cognitive simulation into the maryland virtual patient. *Studies in Health Technology and Informatics*, 142:224–229, 2009. PMID: 19377155.

[171] Uri Nodelman, Christian Shelton, and Daphne Koller. Learning continuous time bayesian networks. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 451–45, San Francisco, CA, 2003. Morgan Kaufmann.

[172] J. B. Olansen, Jr J. W. Clark, D. Khoury, F. Ghorbel, and A. Bidani. A closed-loop

model of the canine cardiovascular system that includes ventricular interaction. *Comput. Biomed. Res.*, 33(4):260–295, 2000.

[173] Rong Pan, Yun Peng, and Zhongli Ding. Belief update in bayesian networks using uncertain evidence. In *Tools with Artificial Intelligence, 2006. ICTAI '06. 18th IEEE International Conference on*, pages 444, 441, December 2006.

[174] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman Publ Inc, May 1997.

[175] Niels Peek, Marion Verduijn, Peter M J Rosseel, Evert de Jonge, and Bas A de Mol. Bayesian networks for multivariate data analysis and prognostic modelling in cardiac surgery. *Studies in Health Technology and Informatics*, 129(Pt 1):596–600, 2007. PMID: 17911786.

[176] J. M. Pena, J. A. Lozano, and P. Larranaga. An improved bayesian structural EM algorithm for learning bayesian networks for clustering. *Pattern Recognition Letters*, 21(8):779–786, July 2000.

[177] PERC. Physiology educational research consortium, http://www.physiologyeducation.org/.

[178] Bruno-Edouard Perrin, Liva Ralaivola, Auralien Mazurie, Samuele Bottani, Jacques Mallet, and Florence d'Alcha-Buc. Gene networks inference using dynamic bayesian networks. *Bioinformatics (Oxford, England)*, 19 Suppl 2:ii138–148, October 2003. PMID: 14534183.

[179] Mikel D. Petty and Piotr S. Windyga. A high level architecture-based medical simulation system. *SIMULATION*, 73(5):281–287, November 1999.

[180] Rex B. Kline PhD. *Principles and Practice of Structural Equation Modeling, Second Edition*. The Guilford Press, second edition edition, September 2004.

[181] J H Philip. Gas man–an example of goal oriented computer-assisted teaching which results in learning. *International Journal of Clinical Monitoring and Computing*, 3(3):165–173, 1986. PMID: 3540162.

[182] Physiome. http://www.physiome.org/.

[183] PhysioNet. The research resource for complex physiological signals, http://www.physionet.org/.

[184] Adam G. Polak and Janusz Mroczka. Nonlinear model for mechanical ventilation of human lungs. *Computers in Biology and Medicine*, 36(1):41–58, 2006.

[185] Pronktech. http://www.pronktech.com.

[186] QCP. Physiological modeling - the university of mississippi medical center, http://physiology.umc.edu/themodelingworkshop.

[187] QT. Toolkit documentation, http://doc.trolltech.com/4.4/.

[188] QT. Toolkit library, http://qt.nokia.com.

[189] Marco Ramoni, Paola Sebastiani, and Paul Cohen. Bayesian clustering by dynamics. *Mach. Learn.*, 47(1):91–121, 2002.

[190] Philipp Rebholz, Carsten Bienek, Dzmitry Stsepankou, and JÃ$\frac{1}{4}$rgen Hesser. CathI a training system for PTCA. a step closer to reality. In *Medical Simulation*, pages 249–255. 2004.

[191] Micaela B. Reddy, R. S. Yang, and Melvin E. Andersen. *Physiologically based pharmacokinetic modeling*. John Wiley and Sons, 2005.

[192] ALDO RESCIGNO. CLEARANCE, TURNOVER TIME, AND VOLUME OF DISTRIBUTION. *Pharmacological Research*, 35(3):189–193, March 1997.

[193] Aldo Rescigno. The birth of compartments. In *Foundations of Pharmacokinetics*, pages 27–35. 2003.

[194] Brian Robinson. USING PATIENT SIMULATORS TO TEACH PHARMACOLOGY AND PATHOPHYSIOLOGY. Technical report, National Patient Simulation Training Centre, New Zealand, anzca, 2004.

[195] Christian Roth. Realtime simulation and modeling of ECG (electrocardiogram) based on parameter estimation from real data sets, diploma thesis, university of mannheim, germany, 2008.

[196] RTI. International - scientific research development institute : http://www.rti.org/.

[197] Jan Rüdiger, Achim Wagner, and Essam Badreddin. Behavior based description of dependability - defining a minium set of attributes for a behavioral description of dependability. In *ICINCO-RA (2)*, pages 341–346, 2007.

[198] Jan Rüdiger, Achim Wagner, and Essam Badreddin. Behavior based dependability estimation - estimating the dependability of autonomous mobile systems using predictive filter. In *ICINCO-RA (2)*, pages 137–142, 2008.

[199] G. W. Rutledge. VentSim: a simulation model of cardiopulmonary physiology. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 878–883, 1994. PMID: 7950050 PMCID: 2247939.

[200] Reza Sameni, Gari D. Clifford, Christian Jutten, and Mohammad B. Shamsollahi. Multichannel ECG and noise modeling: application to maternal and fetal ECG signals. *EURASIP J. Appl. Signal Process.*, 2007(1):94–94, 2007.

[201] Hunor Santha. Medical sensors used for monitoring and diagnosis. *Sensor Review*, 25(4), December 2005.

[202] SBML. http://sbml.org/.

[203] Mark Schmidt, Alexandru Niculescu-Mizil, and Kevin Murphy. Learning graphical model structure using l1-regularization paths. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, pages 1278–1283, Vancouver, British Columbia, Canada, 2007. AAAI Press.

[204] B. Schoeller. Pulsoximeterie - fibel. Technical report, MCC GmbH, 2000.

[205] Howard A. Schwid, G. Alec Rooke, Piotr Michalowski, and Brian K. Ross. Screen-Based anesthesia simulation with debriefing improves performance in a Mannequin-Based anesthesia simulator. *Teaching and Learning in Medicine: An International Journal*, 13(2):92, 2001.

[206] Nirenburg Sergei, Marjorie McShane, and Stephen Beale. A unified ontological-semantic substrate for physiological simulation and cognitive modeling. University at Buffalo, 2009.

[207] A. Shahidi and P. Savard. Forward problem of electrocardiography: construction of human torso models and field calculations using finite element method. *Medical and Biological Engineering and Computing*, 32(0):S25–S33, July 1994.

[208] Tomas Singliar and Denver Dash. Efficient inference in persistent dynamic bayesian networks. In *Proceedings of the Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*, pages 494–502, Corvallis, Oregon, 2008. AUAI Press.

[209] Smith, S Andreassen, Shaw, Jensen, Rees, and Chase. Simulation of cardiovascular system diseases by including the autonomic nervous system into a minimal model. *Comput Methods Programs Biomed*, 86(2):160, 153, May 2007.

[210] Bram W. Smith, J. Geoffrey Chase, Roger I. Nokes, Geoffrey M. Shaw, and Graeme Wake. Minimal haemodynamic system model including ventricular interaction and valve dynamics. *Medical Engineering & Physics*, 26(2):131–139, March 2004.

[211] N Ty Smith and Kenton R Starko. The physiology and pharmacology of growing old, as shown in body simulation. *Studies in Health Technology and Informatics*, 111:488–491, 2005. PMID: 15718784.

[212] Hayssam Soueidan, David Sherman, and Macha Nikolski. Biorica: A multi model description and simulation system. In *Foundations of Systems Biology and Engineering (F0SBE)*, 2007.

[213] Jos A. E. Spaan. *Coronary blood flow*. Springer, 1991.

[214] Harald Steck. *Constraint-Based Structural Learning in Bayesian Networks using Finite Data Sets*. PhD thesis, Department of Informatics, Technical University Munich, May 2001.

[215] Nikos Stergiopulos, Berend E. Westerhof, and Nico Westerhof. Total arterial inertance as the fourth element of the windkessel model. *Am J Physiol Heart Circ Physiol*, 276(1):H81–88, 1999.

[216] James H. Stuhmiller and Louise M. Stuhmiller. A mathematical model of ventilation response to inhaled carbon monoxide. *J Appl Physiol*, 98(6):2033–2044, June 2005.

[217] Subverion. SVN, a version control system, subversion.tigris.org/.

[218] J. Swedenborg. McDonald's blood flow in arteries. *European Journal of Vascular and Endovascular Surgery*, 31(6):679, June 2006.

[219] Robert Lee Sweitzer and Eugene Palatnik. US patent ep164139 - oximetry simulator, April 2006.

[220] T. Teorell. Kinetics of distribution of substances administered to the body. *Arch Int Pharmacodyn Ther*, 57:205–225, 1937.

[221] Kamlesh M. Thakker. Compartmental models and their application, keith godfrey. academic press inc., london, 1983. no. of pages: 293, price: $50.00. *Biopharmaceutics & Drug Disposition*, 6(3):357–358, 1985.

[222] W. James Thoman, Dietrich Gravenstein, Jan van der Aa, and Samsun Lampotang. Autoregulation in a Simulator-Based educational model of intracranial physiology. *Journal of Clinical Monitoring and Computing*, 15(7):481–491, December 1999.

[223] S Thomas, Enas Abdulhay, Pierre Baconnier, Julie Fontecave, Jean-Pierre Francoise, Francois Guillaud, Patrick Hannaert, Alfredo Hernandez, Virginie Le Rolle, Pierre Maziere, Fariza Tahi, and Farida Zehraoui. SAPHIR - a multi-scale, multi-resolution modeling environment targeting blood pressure regulation and fluid homeostasis. *Conference Proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2007:6649–6652, 2007. PMID: 18003550.

[224] Peter Thygesen, Panos Macheras, and Achiel Van Peer. Physiologically-Based PK/PD modelling of therapeutic macromolecules. *Pharmaceutical Research*, 26(12):2543–2550, December 2009.

[225] TianRong. Tianrong medical equipment, www.tianrong.cn.

[226] Harukazu Tsuruta, Toshiro Sato, Masuo Shirataka, and Noriaki Ikeda. Mathematical model of cardiovascular mechanics for diagnostic analysis and treatment of heart failure: Part 1 model description and theoretical analysis. *Medical and Biological Engineering and Computing*, 32(1):3–11, 1994.

[227] Mauro Ursino. Interaction between carotid baroregulation and the pulsating heart: a mathematical model. *Am J Physiol Heart Circ Physiol*, 275(5):H1733–1747, November 1998.

[228] Claudia Valdez-Lowe, Sameh A. Ghareeb, and Nancy T. Artinian. Pulse oximetry in adults. *AJN The American Journal of Nursing*, 109(6), 2009.

[229] VAM. Virtual anesthesia machine: http://vam.anest.ufl.edu/.

[230] Marcel A J van Gerven, Babs G Taal, and Peter J F Lucas. Dynamic bayesian networks as prognostic models for clinical patient management. *Journal of Biomedical Informatics*, 41(4):515–529, August 2008. PMID: 18337188.

[231] Willem L. van Meurs and Tammy Y. Euliano. Model driven simulators from the clinical instructor's perspective: current status and evolving concepts. In *Proceedings of the second annual conference on Simulators in anesthesiology education*, pages 65–73, Rochester, New York, United States, 1998. Plenum Press.

[232] Willem L. van Meurs, Michael L. Good, and Samsun Lampotang. Functional anatomy of Full-Scale patient simulators. *Journal of Clinical Monitoring and Computing*, 13(5):317–324, 1997.

[233] A van Oosterom. Genesis of the t wave as based on an equivalent surface source model. *Journal of Electrocardiology*, 34 Suppl:217–227, 2001. PMID: 11781960.

[234] A van Oosterom. ECGSIM: an interactive tool for studying the genesis of QRST waveforms. *Heart*, 90(2):165–168, 2004.

[235] Matthias Veit and Stephan Herrmann. Model-view-controller and object teams: a perfect match of paradigms. In *Proceedings of the 2nd international conference on Aspect-oriented software development*, pages 140–149, Boston, Massachusetts, 2003. ACM.

[236] Marion Verduijn, Niels Peek, Peter M.J. Rosseel, Evert de Jonge, and Bas A.J.M. de Mol. Prognostic bayesian networks: I: Rationale, learning procedure, and clinical use. *Journal of Biomedical Informatics*, 40(6):609–618, December 2007.

[237] Marion Verduijn, Peter M.J. Rosseel, Niels Peek, Evert de Jonge, and Bas A.J.M. de Mol. Prognostic bayesian networks: II: an application in the domain of cardiac surgery. *Journal of Biomedical Informatics*, 40(6):619–630, December 2007.

[238] H Wasyluk, A Onisko, and M J Druzdzel. Support of diagnosis of liver disorders based on a causal bayesian network model. *Medical Science Monitor: International Medical Journal of Experimental and Clinical Research*, 7 Suppl 1:327–332, May 2001. PMID: 12211748.

[239] Web-Human. Physiology teaching simulation : http://placid.skidmore.edu.

[240] Philippe Weber and Lionel Jouffe. Complex system reliability modelling with dynamic object oriented bayesian networks (DOOBN). *Reliability Engineering & System Safety*, 91(2):149–162, February 2006.

[241] John G. Webster. *Design of pulse oximeters*. CRC Press, 1997.

[242] Sholom M. Weiss, Casimir A. Kulikowski, Saul Amarel, and Aran Safir. A model-based method for computer-aided medical decision-making. *Artificial Intelligence*, 11(1-2):145–172, August 1978.

[243] Nico Westerhof. Law of LaPlace. In *Snapshots of Hemodynamics*, pages 31–33. 2005.

[244] Dan Wu and Cory Butz. On the complexity of probabilistic inference in singly connected bayesian networks. In *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, pages 581–590. 2005.

[245] R S Yang, R S Thomas, D L Gustafson, J Campain, S A Benjamin, H J Verhaar, and M M Mumtaz. Approaches to developing alternative and predictive toxicology based on PBPK/PD and QSAR modeling. *Environmental Health Perspectives*, 106(Suppl 6):1385–1393, December 1998. PMID: 9860897 PMCID: 1533423.

[246] D. Zinemanas, R. Beyar, and S. Sideman. An integrated model of LV muscle mechanics, coronary flow, and fluid and mass transport. *Am J Physiol Heart Circ Physiol*, 268(2):H633–645, February 1995.

[247] Geoffrey Zweig and Stuart Russell. Speech recognition with dynamic bayesian networks. pages 173–180, 1998.

12 Appendix

12.1 Rapid Hardware Development

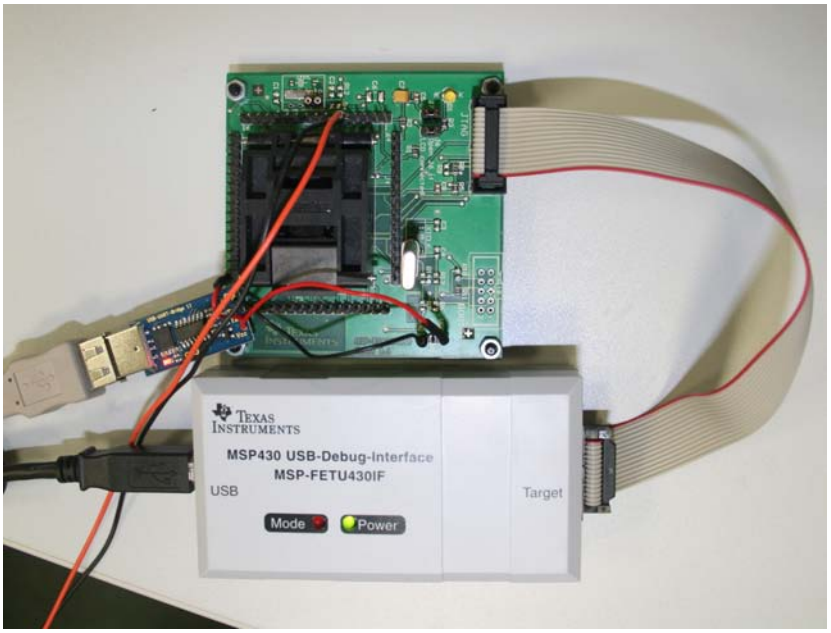


Figure 12.1: MSP430FET development board: supports a standalone ZIF socket target board (64-pin LQFP package) used to program (flash emulation tool (FET)) and debug the MSP430 μC in-system through the JTAG interface or the Spy Bi-Wire (2-wire JTAG) protocol.

12.2 PCB Desgin

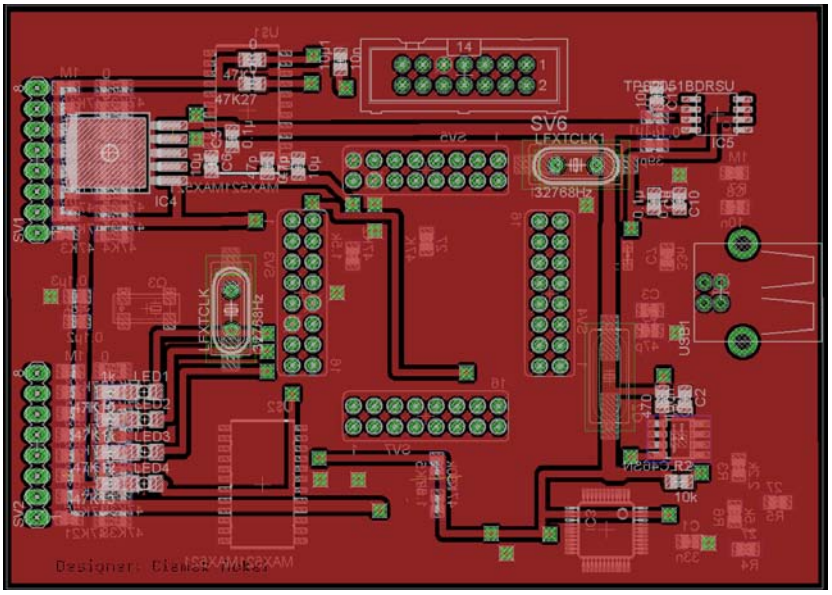


Figure 12.2: ECG emulation PCB top.

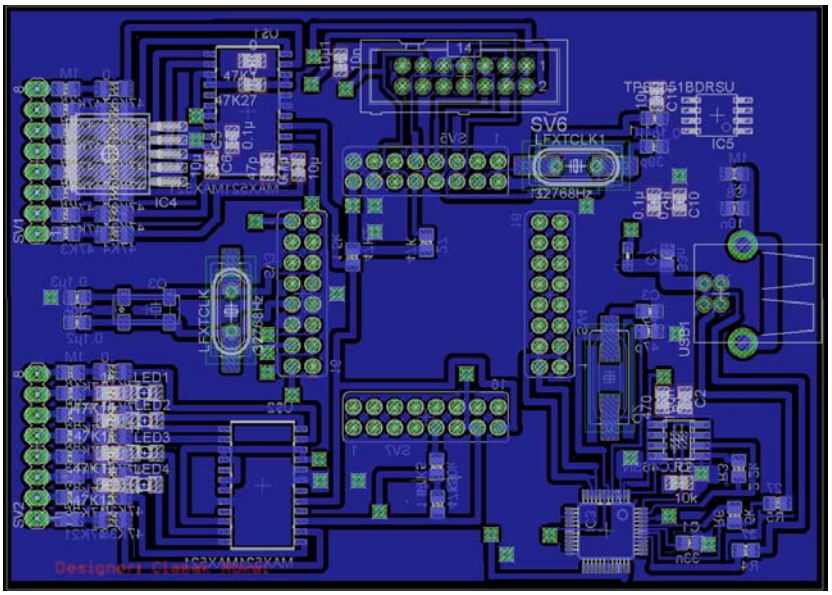


Figure 12.3: ECG emulation PCB bottom.

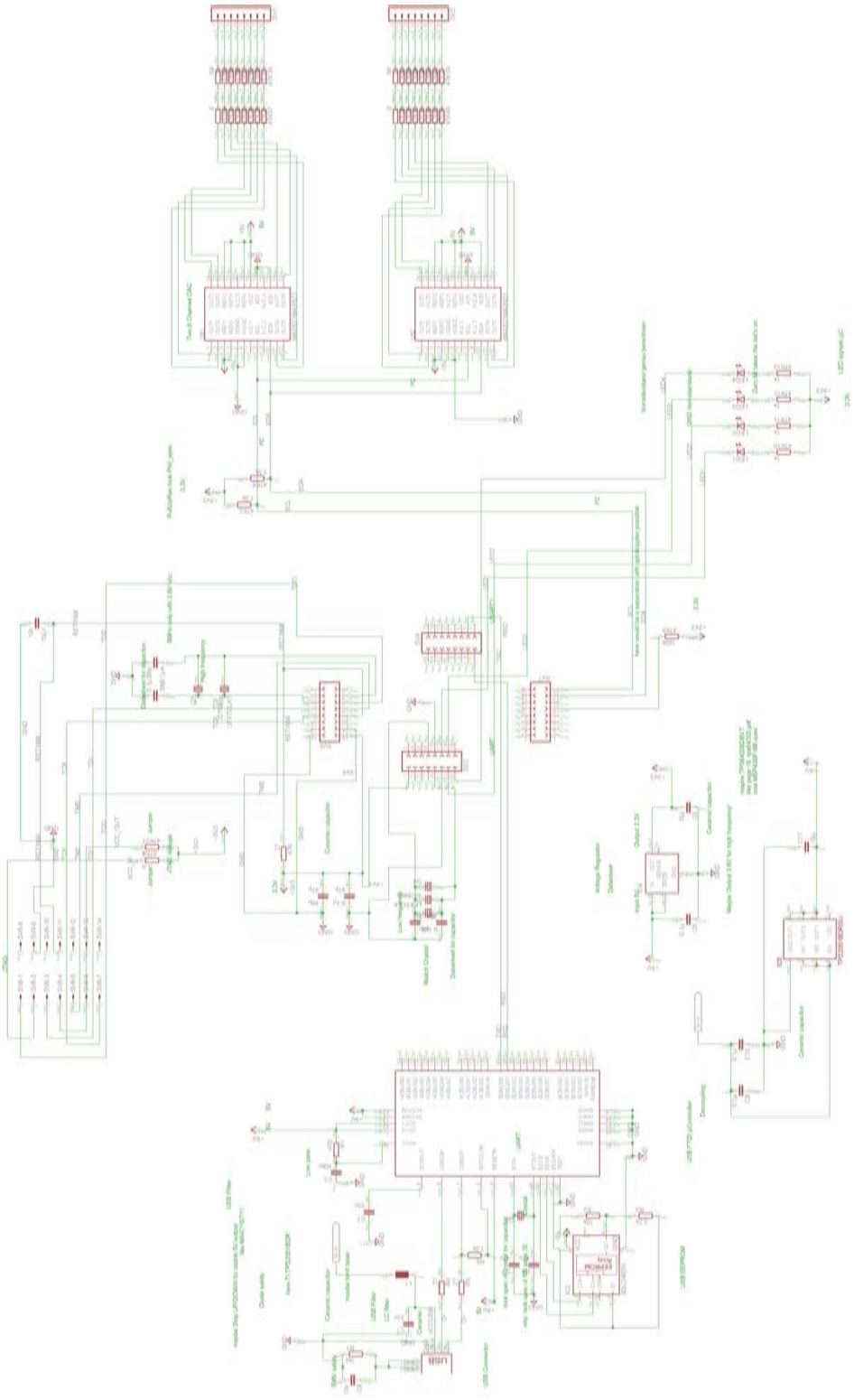


Figure 12.4: ECG emulation PCB schematic.

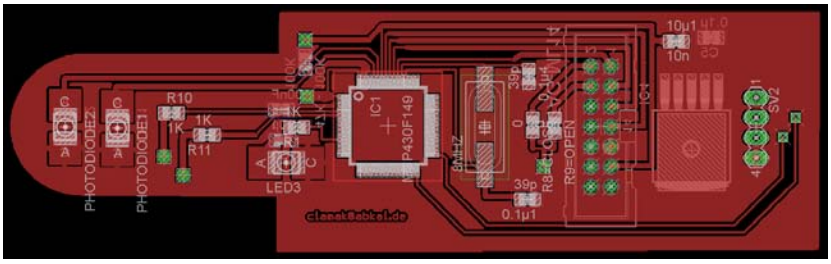


Figure 12.5: Pulseoximetry emulation PCB layout top.

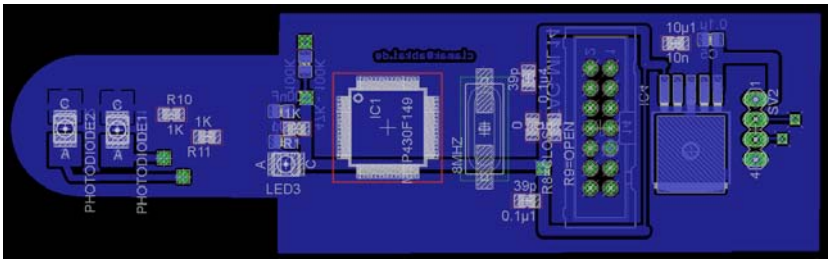


Figure 12.6: Pulseoximetry emulation PCB layout bottom.

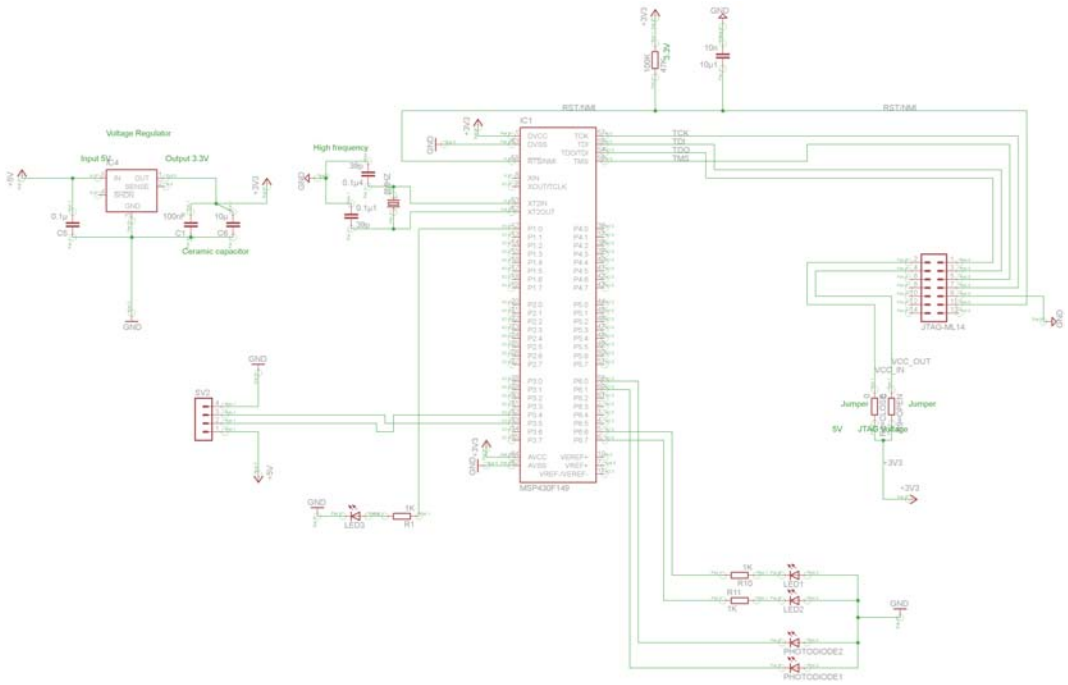


Figure 12.7: Pulseoximetry emulation PCB schematic.

12.3 Remote Monitoring

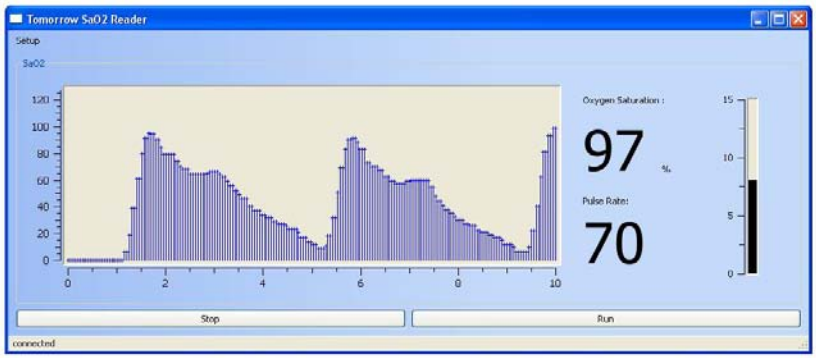


Figure 12.8: SaO2 special purpose remote monitoring software. This software was specially designed for remote digital recording of SaO2 vital signal parameters, e.g. HR, saturation and waveform, for later evaluation and processing.

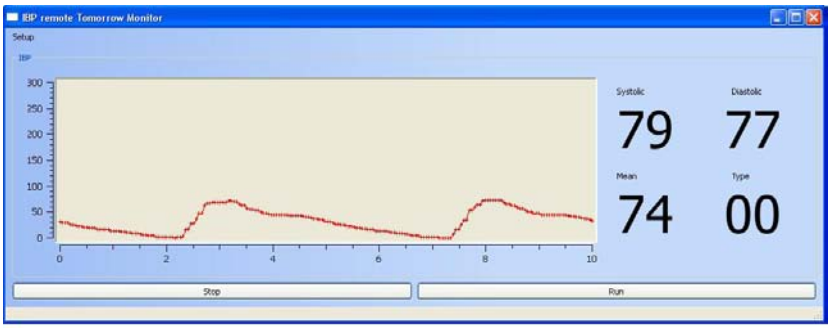


Figure 12.9: IBP special purpose remote monitoring software. This software was specially designed for remote digital recording of IBP pulsative signal, including curve, mean, systolic and diastolic values.