

# Towards Medical Cyber-Physical Systems: Modelica and FMI based Online Parameter Identification of the Cardiovascular System

Jonas Gesenhues<sup>1</sup> Marc Hein<sup>2</sup> Maike Ketelhut<sup>1</sup> Thivaharan Albin<sup>1</sup> Dirk Abel<sup>1</sup>

<sup>1</sup>Institute of Automatic Control, RWTH Aachen University, Germany [j.gesenhues@irt.rwth-aachen.de](mailto:j.gesenhues@irt.rwth-aachen.de)

<sup>2</sup>Department of Anesthesiology, RWTH Aachen University Hospital, Germany

## Abstract

This paper presents a concept for online parameter identification intended to be used within cardiovascular research labs and hospitals of the future featuring a data network of medical sensors. It is based on iterative non-linear optimization using a moving horizon scheme and object-oriented *Modelica* models. Special FMUs have been developed to interface the optimization module and the sensor hardware. The concept is demonstrated on an exemplary application of identifying the parameters of a model for the systemic circulation. Unlike classical online parameter identification methods, this concept allows for quickly implementing changes of the underlying model. *Keywords: Online Parameter Identification, Moving Horizon, FMI, ModeliChart, JModelica.org, CasADi, Cardiovascular, Medical*

## 1 Introduction

Throughout many countries around the globe, public health care systems are being faced by the ongoing trend of increased demand for health care services. On the one hand, this is due to the consequences of demographic changes towards an aging population. On the other hand, scientific progress allows for increased treatment possibilities (European Commission, 2016). At the same time, public hospitals, a major pillar within the health care systems, are faced by a lack of qualified health care personal. Supporting health care personal in public hospitals by smart technology might provide an essential component to meet those challenges. In this regard, ongoing trends such as digitalization of information, large scale data agglomeration ('*Big Data*'), interconnection of devices ('*IoT*') and smart algorithms that allow for e.g. automated monitoring of a patient's status and early recognizing and possibly automatically resolving critical conditions can be expected to find their way into hospitals in the future and have the potential to improve the outcome of patients.

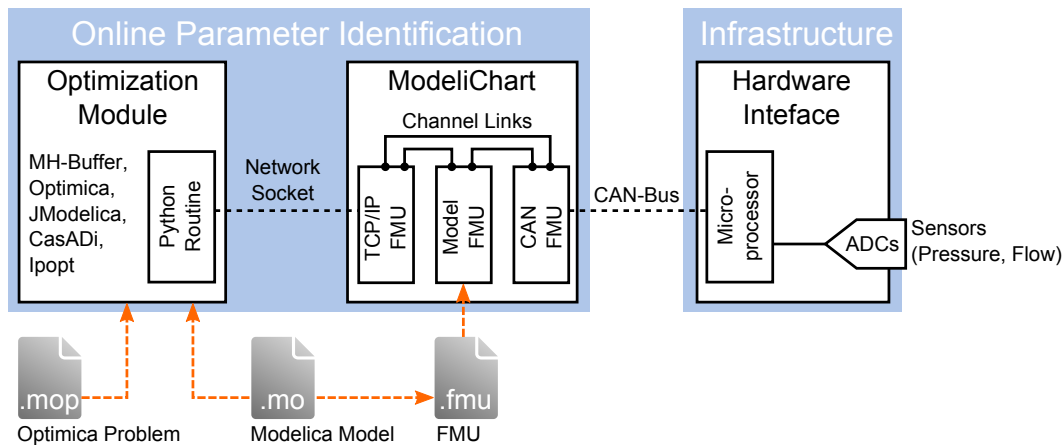
Within this context, the research focus of our interdisciplinary group consisting of engineers and physicians is on improving the therapy of terminal heart failure, the most prevalent cause of death in the western world (Nichols et al., 2012). Specifically, we are working on control

strategies for technological heart assist devices, such as blood pumps that are connected to the body to assist the heart (Ventricular Assist Devices). Here, mathematical models of the cardiovascular system are applied in many different ways, ranging from computer '*model in the loop*' simulations of new control strategies (e.g. Habigt et al. (2016); Ketelhut et al. (2017)) over driving test benches for '*hardware and software in the loop*' hardware tests (e.g. Misgeld et al. (2015)) to state estimation (e.g. Rüschen et al. (2016)) and model based control (Gesenhues et al., 2016).

Although much literature exists describing the observed behavior of the healthy body, few is known about the underlying mechanisms and how they are affected by diseases, drugs or the interaction with technical devices. Consequently, the adaption, refinement and creation of new models is an integral part within this field. Here, over the years the object-oriented modeling paradigm using *Modelica* has turned out invaluable for its flexibility for modifications and the concept of acausal formulation of components (Gesenhues et al., 2017) and has motivated the creation of libraries such as the *Physiolibrary* (Matejak et al., 2014) or our in-house developed library *HumanLib* (Brunberg et al., 2009).

Besides model structure, the identification of the contained model parameters is important. When it comes to biomedical dynamical systems such as the cardiovascular system, there is generally a great extent of variation considering parameters. First of all, parameters vary with countless individual characteristics of patients such as age, height, weight, gender, lifestyle etc. Second, the presence and extend of diseases directly affects the parameters. Finally, even in a specific single patient at a specific state, the parameters vary because the body possesses numerous physiological control mechanisms to adapt to external conditions such as temperature, exercise or even the current posture (standing upright or lying). Thus, the model parameters need to be considered time varying and can change within seconds. All in all, parameters identified from measured data represent a snapshot of an individual patient at a specific time.

For all of those reasons and having smart algorithms and features of hospitals in the future in mind, an au-



**Figure 1.** Architecture and components of the online identification concept. MH: Moving Horizon, CAN: Control Area Network, ADC: Analog-Digital Converter.

tomated online model parameter identification procedure that is capable to be included in a hospital’s data network and which continuously identifies model parameters based on live patient data provides many benefits including diagnostic assistance to doctors (model parameters can be used to assess a patient’s status), smart alarms that are raised when selected parameters exceed a certain threshold up to model based control of medical devices, for which an adequately parametrized model is an essential prerequisite. In-vivo animal trials are an integral element of our research. The infrastructure that we have developed to conduct such trials features a large number of sensors which are connected to a data network (currently we are using the Control Area Network (CAN) bus). This infrastructure bears resemblance to the possible infrastructure of future hospitals. Thus, our trials and infrastructure provide a test bed for the implementation of medical cyber-physical systems.

The current state of technology includes many classical online and offline parameter identification methods, which have been adapted and applied to virtually any physical domain. Specifically for the cardiovascular system, those include attempts using reformulation of model equations to allow for (recursive) least square techniques (Clark et al., 1980; Hann et al., 2006; Kosaka et al., 2002) and Bayesian approaches like the (extended) Kalman filter (Yu et al., 1998). Although it has been shown that the results yielded from classical approaches are valid and reliable, a major limitation consists in the fact that there is an enormous effort to reformulate the model into the specific form needed for the identification method. This generally includes manually rearranging equations and to transform the system by introducing new state variables and parameters (e.g. to resolve non-linearities). Shortly, classical methods might be satisfactory when the underlying model meets the requirements of the identification method and can be considered ‘frozen’ with the start of development as revisions to the model at a later time can be laborious and even impossible to implement.

As mentioned above, cardiovascular system models are subject to frequent changes. Thus, the applicability of classical methods is limited in this regard. Those limitations motivate new online identification procedures which do not require excessive reformulation efforts of the underlying model. Recently, we have started to consider non-linear optimization based identification using our *Modelica* models in combination with *Optimica*, *JModelica.org* and *CasADi* as a possible solution. A recent study focusing on the offline identification of patient specific parameters using those tools comes to the conclusion that patient specific parameter identification has the potential to be a promising component for patient assessment in the clinic (Moza et al., 2017).

The contribution of this paper is a concept that allows for the automated online parameter identification based on those ideas and tools which does not exhibit the described limitations of the previous state of the art and can be used within our animal trial infrastructure. The general idea is to repeatedly (re-)identify the current parameter values by solving a non linear optimization problem over a short time interval. The paper is organized as follows: first, the next section provides a general overview over the concept and its components and the typical setup work flow involved. Section 3 details the iteratively carried out optimization procedure. Afterwards, the concept is demonstrated by the exemplary application of identifying the parameters involved in a simple model of the systemic circulation (Section 4). Finally, the results are presented and a discussion on current limitations and further enhancements is given (Section 5).

## 2 Concept Overview and Work Flow

The components involved within the presented concept are summarized in Figure 1. The concept consists of our FMU-master *ModeliChart* (see Section 2.2 below) which serves as the central hub and graphical user interface and of the optimization module, which constantly calculates

current parameter values. It is further described in Section 3. The optimization module is implemented as a Python routine. Data exchange between those two components is realized through a TCP socket stream using a simple custom protocol. This design allows for either running the optimization module on the same machine that is running *ModeliChart* as well as running the optimization module on a different machine (possibly outside of the lab) connected to the local area network (LAN).

## 2.1 Interface FMUs

To realize the architecture depicted in Figure 1 two FMUs complying with the FMI 2.0 co-simulation standard have been developed to allow for data exchange between the individual components of the concept. Both interface FMUs have in common that they are fully configurable through the `modelDescription.xml` and additional configuration files provided as resources. So far, both FMUs only support the `Real` data type. An arbitrary number of input and output channels that appear as scalar FMU variables can be set. Input channels are intended for receiving data, output channels, which are marked with the attributes `causality="parameter"` and `variability="tunable"` are intended to send data. For each of both FMUs a convenient software tool has been developed to automatically generate the according `modelDescription.xml`, additional configuration files and the packed FMU.

The first FMU constitutes a TCP/IP based network socket interface used for the connection between the optimization module and *ModeliChart*. During the initialization of the FMU, a TCP server accepting connections on a configurable port is started waiting for a client (here the optimization module) to connect. At this point, only a single client is supported. On every execution of the `doStep(...)` method, the values of the output channel scalar variables are sent to the connected client using a simple custom protocol. Similarly, the `getReal(...)` function returns the latest value of the specified input channel scalar variable. The client is allowed to send values at any given time.

The second FMU allows for the interaction with the CAN bus of our infrastructure which distributes the sensor signals. This FMU uses the API provided by the manufacturer of the CAN interface hardware (PEAK-System Technik GmbH, Darmstadt, Germany). The CAN FMU listens to CAN messages of preconfigured message identifiers and returns the last received value whenever the corresponding `getReal(...)` function is called. Although not required in the here presented application, the CAN FMU also supports sending values to the CAN bus.

## 2.2 ModeliChart

*ModeliChart* is our self-developed freely available FMU host. The original motivation has been to provide a free and intuitive opportunity to assess and play with simulation models to physicians. However, the ease of use and

the hardware interaction capabilities through the interface FMUs described above have turned *ModeliChart* into a 'Swiss army knife' for all steps during rapid control prototyping cycles. Based on the .NET framework (Microsoft, Redmond, WA, US), it provides a simple intuitive graphical user interface. *ModeliChart* supports FMUs complying with the FMI 2.0 co-simulation standard. The main intended use case is real time operation by periodically calling the `doStep(...)` method of all FMUs after a configurable time interval. So called 'channel links' allow individual FMUs to be connected: Internally, for each channel link the `SetReal(...)` method of the receiving FMU is called at each time step. More details on *ModeliChart* can be found in (Gesenhues et al., 2017).

Within the here presented application, three FMUs are used. The CAN FMU is used to fetch the measurements of the sensors of interest from the CAN bus. Through channel links, the measurement data is handed to the TCP/IP FMU which in turn sends the measurement data to the optimization module. In this regard *ModeliChart* serves as a CAN to TCP/IP bridge. After new identified parameters are available from the optimization module, they are sent to *ModeliChart* through the TCP/IP FMU. The third FMU contains the model under investigation. Through channel links the current parameter values are set to the model FMU. In combination with the measurement data from the CAN FMU used as input into the model FMU, it is possible to compare chosen simulated signals with corresponding measured signals. Each channel can be plotted allowing to observe trends of parameter values and to visually assess the validity of the results by comparing the simulated and measured signals.

## 2.3 Typical Setup Work Flow

Typically, the setup starts with a *Modelica* model that contains the parameters of interest. The model should contain variables that correspond to measured signals. Obviously it is required that the parameters are adequately related to the variables representing the measured signals (i.e. observability, correlation etc.). Next, an *Optimica* file is created containing the optimization problem. The `optimization` class should extend from the *Modelica* model. In simple cases, it can be sufficient to just mark the parameters to be identified as 'free' when a quadratic penalty cost function is to be used. Nevertheless it is also possible to define custom cost functions. For consistency, certain variable expressions for optimization parameters such as final time or limits on allowed parameter values can be used which are later overwritten when the *Optimica* file is loaded in the optimization module. An exemplary *Optimica* file can be seen in Listing 2. Both files are placed in a folder accessible for the optimization module. Next, some adaption of the *Python* routine is necessary to match the measurement data to the corresponding model variables and to set optimization parameter settings (see Section 3). Besides, an FMU of the *Modelica* model to be used within *ModeliChart* is created and if necessary the

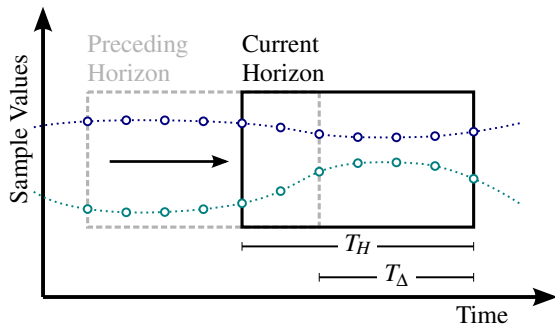


Figure 2. Illustration of the moving horizon scheme.

configuration of the interface FMUs is adapted. Finally, a *ModeliChart* setup (loading the FMUs and creating the channel links) is created and optionally saved for later use. As soon as the setup has been loaded in *ModeliChart*, the optimization module can be started.

### 3 Optimization Module

The online parameter identification is realized through continuously iteratively solving the optimization problem defined within the *Optimica* problem using the measurement data samples within a most recent finite time frame (horizon). Using a moving horizon scheme (Figure 2), as soon as enough samples for a new horizon spanning the time  $T_H$  and enough time since the preceding horizon  $T_\Delta$  has passed, a new optimization job on the current horizon measurement data is dispatched.

The optimization module has been implemented as a Python routine. It uses the modules provided by *JModelica.org* (version 1.17) and in particular the integrated *CasADi* based optimization tool chain (Åkesson et al., 2010; Andersson et al., 2011). *CasADi* is a nonlinear optimization framework that is capable to automatically discretize the optimization problem using a collocation scheme and to calculate the necessary derivatives through algorithmic differentiation. The tool chain automatically transforms the formulated *Optimica* problem to be solved by a non linear optimization solver. Here, *Ipopt* has been used (Wächter and Biegler, 2005).

Figure 3 provides an overview of the routine. After the optimization module has been started, the *Modelica* and *Optimica* files are loaded. The variable expressions in the *Optimica* file (see Listing 2) are replaced by the configured values within the routine, i.e. `%FINAL_TIME%` is set to the value  $T_H$ . Using the according modules, the *Modelica* model is simulated with artificial input signals over the time frame  $T_H$  to provide initial trajectories of all variables. Afterwards, the optimization problem is compiled and discretized using the `prepare_optimization(...)` function of the `transfer_optimization_problem` module. The so prepared discretized problem will be used for each of the following optimizations. Since this compilation process takes significantly longer than the actual solution of the

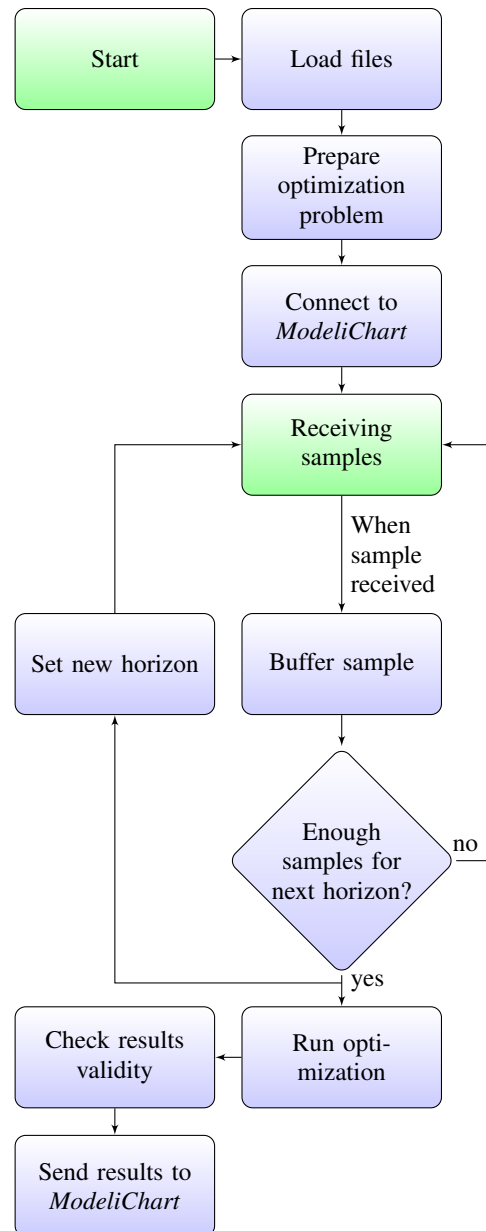


Figure 3. Flow diagram of the optimization module routine.

optimization problem, reusing the prepared discretization saves a lot of computation time. Afterwards, as soon the connection to *ModeliChart* is established, incoming samples are awaited and buffered. As soon as a new horizon is collected according to the described moving horizon scheme, the samples within the horizon are set as the new external data. Furthermore, the initial trajectories for the solution of the optimization problem are set to the solution trajectories of the preceding optimization. Afterwards the solver is started.

For a number of reasons depending on the application (some examples will be shown in Section 4), the solution obtained from the solver might be invalid or the solver might even fail to find a solution within a reasonable time. At this point, it is just checked whether the found parameters are within defined limits to decide on the validity of

**Listing 1.** *Modelica* model for the identification of the TEW parameters. The associated *Modelica* library 'HumanLib' can be found as online supplement.

```

model Ident3ElemWK
  import HumanLib.Basics.*
  import HumanLib.Vessels.*
  parameter Real param_Z=0.1;
  parameter Real param_R=1;
  parameter Real param_C=1;
  Resistance Z(R=param_Z);
  Compliance C(V_0=0, C=param_C,
    V(start=100, fixed=false));
  Resistance R(R=param_R);
  Sources.PressureSource_Variable P_Ao;
  Sources.PressureSource_Variable P_CV;
  input Real AoP;
  input Real CVP;
  Sensors.FlowSensor Q_Ao;
equation
  connect (Z.cnStreamOut, R.cnStreamIn);
  connect (C.cnBloodStream, R.cnStreamIn);
  connect (R.cnStreamOut, P_CVP.cnBloodStream
    );
  connect (P_Ao.P, AoP);
  connect (P_CV.P, CVP);
  connect (Z.cnStreamIn, Q_Ao.cnStreamOut);
  connect (P_Ao.cnBloodStream,
    Q_Ao.cnStreamIn);
end Ident3ElemWK;
    
```

**Listing 2.** *Optimica* optimization problem for the identification of the TEW parameters.

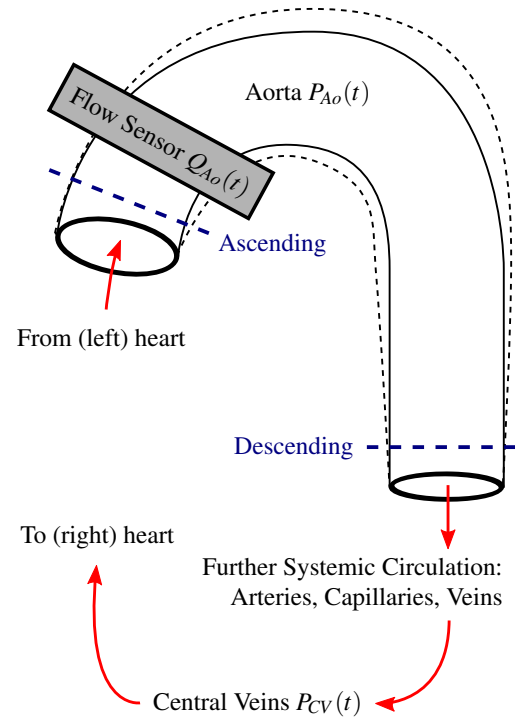
```

optimization OptimizeWKParams(startTime=0,
  finalTime=%FINAL_TIME%)
extends Ident3ElemWK(
  param_Z(free=true,min=%MIN_Z%,max=%MAX_Z%),
  param_R(free=true,min=%MIN_R%,max=%MAX_R%),
  param_C(free=true,min=%MIN_C%,max=%MAX_C%)
  );
end OptimizeWKParams;
    
```

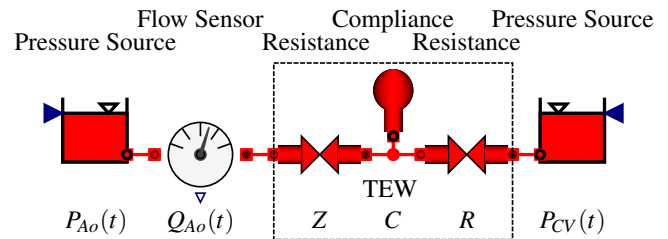
the results but in the future it might make sense to evaluate other criteria like residuals, solution time etc. Therefore, an additional parameter is reported back indicating whether the result is considered valid. This provides feedback for the user on which it can be decided to just ignore 'sporadically' invalid results or to investigate the reason.

## 4 Exemplary Application: Identification of the Systemic Circulation

The concept is demonstrated on the simple but relevant in practice use case of identifying the parameters involved in modeling the flow dynamics of the systemic circulation. The systemic circulation refers to all blood vessels (arteries, capillaries, veins) between the outlet of the left (side of the) heart, which pumps the blood into the systemic circulation and the right heart, which pumps blood into the pulmonary (lung) circulation (Figure 4). The vessels



**Figure 4.** Schematic overview of the systemic circulation.



**Figure 5.** Graphical representation of Listing 1: the three-element-windkessel (TEW) in model combination with additional components for the parameter identification process.

within the systemic circulation start with the arteries and branch up more and more into smaller vessels (ultimately into so called capillaries) running through all parts of the body (muscles, organs) except the lungs. The capillaries end up in the veins, which in turn ultimately end in the big central veins and finally in the right heart.

The large arteries, most importantly the aorta are elastic and have the ability to distend with raising blood pressure and recoil with falling blood pressure (indicated through the dashed line in Figure 4). This leads to a damping of the amplitude of the pulsating blood pressure wave coming from the heart, an effect that is commonly referred to as the physiological windkessel effect. A very simple model for the systemic circulation is the three-element-windkessel (TEW) model (Westerhof et al., 2009). It consists of two hydraulic resistances ( $\Delta P = R \cdot Q$ , with  $\Delta P$  being the blood pressure difference across the element,  $Q$  the blood flow and  $R$  the resistance parameter) and a compliance element ( $V = C \cdot P$  with  $V$  being the current blood volume inside the element,  $P$  the current blood pressure inside the element and  $C$  being the compliance parame-

ter). For convenience, the elements and parameters will be referred to as  $Z, R$  (resistances) and  $C$  (compliance). A graphical representation of the model can be seen in Figure 5.

During a typical in-vivo animal trial on an anesthetized pig, the chest is opened and the heart and the aorta uncovered. This allows for the placement of various sensors. For this application two invasive pressure sensors are used that are placed inside the Aorta ( $P_{Ao}(t)$ ) and inside one of the central veins ( $P_{CV}(t)$ ). Furthermore, an ultrasonic flow sensor is placed at the beginning of the Aorta ( $Q_{Ao}$ , see Figure 4). Listing 1 contains the complete *Modelica* model based on components from our library 'HumanLib' (Brunberg et al., 2009). The blood connector used in Listing 1 consists of the potential variable blood pressure (traditionally denoted in mmHg where  $1 \text{ mmHg} = 133.3 \text{ Pa}$ , referred to atmospheric pressure) and the flow variable blood flow in ml/sec.

Listing 2 contains the according optimization problem. In this example the parameters to be identified are marked free. The optimization problem consists in finding those parameter values that minimize the quadratic difference between the measured signals and the 'simulated' signals. For this, the `ExternalData` class of the `pyjmi.optimization.casadi_collocation` module has been used. It allows to optimize for all three sensor signals at the same time without the need for deciding which signals are considered as input or output signals. The extend of individual signal differences can be weighted against each other. Here, the parameters given in Table 1 have been used to roughly normalize the signals.

To influence the parameters all sorts of experiments can be performed during an animal trial. Here, we consider the constriction of the aorta using a surgical band at two different positions: at the ascending part right at the beginning of the aorta and at the descending part of the aorta as indicated by the dashed lines in Figure 4. It is important to note that the pressure sensor measuring  $P_{Ao}$  measures the aortic pressure right behind the constriction of the ascending position but way before the descending position. Each of those constriction positions should have a different impact on the model parameters.

## 5 Results and Discussion

For the results presented in this section, already available raw data recorded during animal experiments conducted on anesthetized pigs (approved by local animal care authorities) has been used. There were no animal trials conducted for this study. Accordingly, the infrastructure has been emulated to generate the results presented in this section. The raw data contained the sampled values of the sensors at a sample rate of 1 kHz. The settings that have been used are summarized in Table 1.

Two different experiments are investigated. The first experiment is a short constriction of about 20 seconds

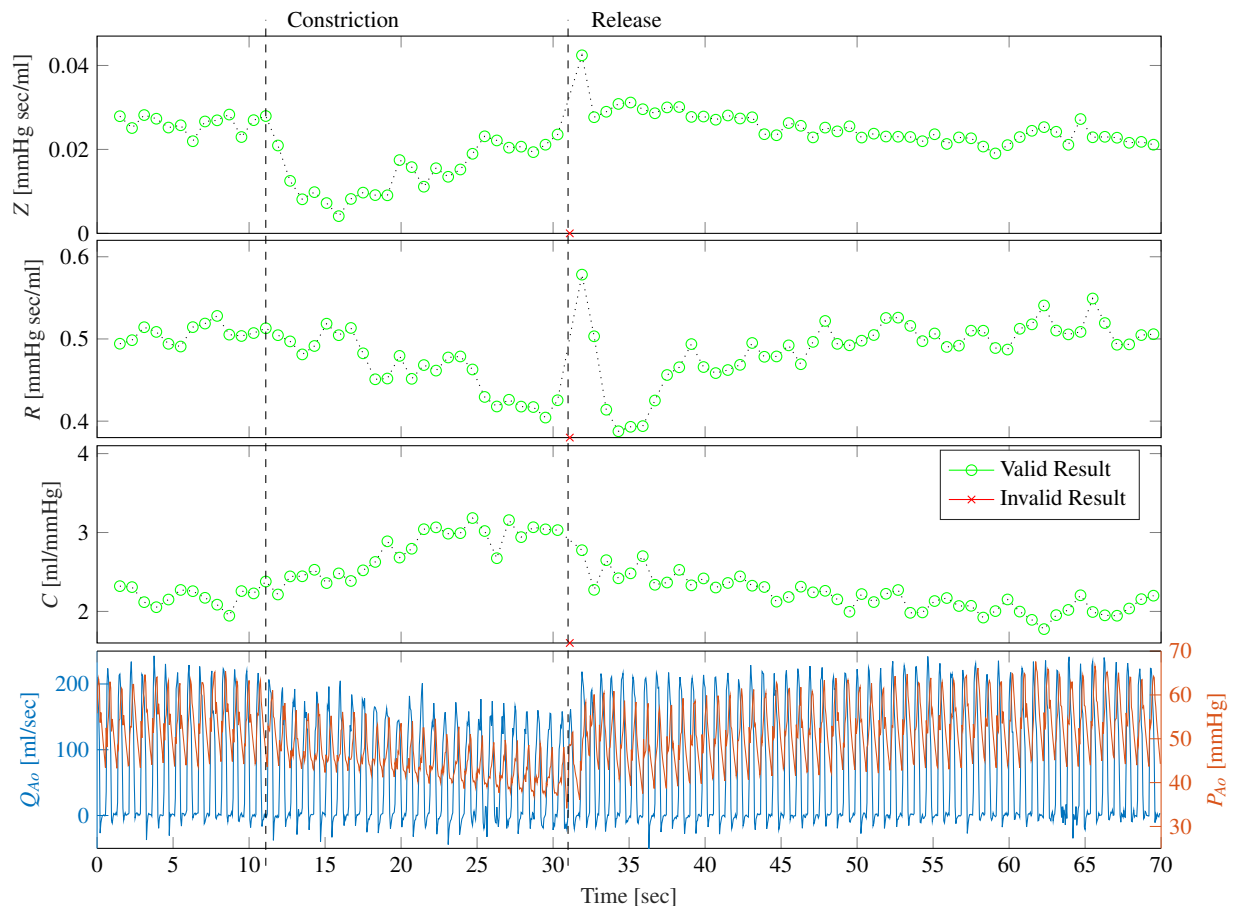
**Table 1.** Settings that have been used to obtain the presented results. All other settings have been left at their default values.

Name	Value
<b>General:</b>	
Horizon length $T_H, \%FINAL\_TIME\%$	1.5 sec
Time between horizons $T_\Delta$	0.8 sec
<b>Validity criteria:</b>	
Maximal value $\%MAX\_R\%, \%MAX\_C\%$	3.5
Maximal value $\%MAX\_Z\%$	1
Minimal value $\%MIN\_R\%, \%MIN\_C\%$	0.1
Minimal value $\%MIN\_Z\%$	0.001
<b>Quadratic penalty weight factors:</b>	
For $P_{Ao}$	10
For $P_{CV}$	20
For $Q_{Ao}$	1
<b>Optimization settings:</b>	
Number of collocation elements	23
Max. <i>Ipopt</i> iterations	300

at the ascending position (Figure 6). The  $P_{CV}$  signal is not shown in the plots since it remains almost constant at around 15 mmHg during the experiments. The pressure  $P_{Ao}$  is measured behind the ascending occlusion position. Hence, constricting the aorta at the ascending position limits the blood flow but hardly affects the properties of the systemic circulation. It can be seen that the parameters change only slightly during the constriction and return to their initial values some time after releasing the constriction. The immediate parameter value changes (most notably the sudden decrease of  $Z$ ) can be explained by nonlinearities of the real system which become apparent when the blood flow is significantly reduced. The slow changes of parameters after the constriction are due to reactions of the bodies regulation mechanisms; trough muscle cells within the wall of some of the arteries the cross section area of the vessel and thus the resistance of the vessel can be controlled by the body. The increase in  $Z$  can be explained by the aim of the body to increase the pressure in the aorta (the so called baroreceptor reflex). Similar, the reduced supply of oxygen (hypoxia) leads to a widening of the blood vessels to allow for increased blood flow and results in a reduction of  $R$ . After the release, the regulatory mechanisms slowly revert the parameters back to the original values.

For the second experiment, a constriction at the descending position for several minutes is performed (Figure 7, release not shown in the figure). When the aorta is constricted at the descending position, the overall resistance of the systemic circulation is drastically increased which is reflected in a significant increase of  $R$ . However, due to impaired draining a significant expansion of the aorta results in a reduction of  $Z$  due to the increase of the cross section area of the aorta. For the same non-linearity reasons as in the first experiment, the expanded aorta exhibits a reduce elastance. Hence, the value of the compliance





**Figure 6.** Short constriction of the aorta at the ascending position with subsequent release after about 20 seconds.

$C$  decreases. Again, although small compared to the effects of the constriction, slight changes of the parameters can be observed after the constriction due to the regulatory mechanisms described above.

As seen in both experiments, the assumed proportionality between pressure and volume to model the compliance  $C$  is only valid within a limited range that has been exceeded during the experiments. Based on this findings it can be considered to revise the model accordingly. Here, the major advantage of the here presented concept consists in the fact that such model adaption can be implemented quickly.

The limitations considering what can be modeled are mostly determined by the features yet supported by *JModelica.org*, *CasADi* and *Ipop*. A major limitation is the requirement of the model equations to be twice continuously differentiable (Wächter and Biegler, 2005) for each optimization variable. This prohibits the use of switching components that commonly include `if...else` statements. In our applications, this affects models containing heart valves. To work around this limitation continuous approximations have been used (Gesenhues et al., 2016, 2017).

On a standard personal computer, the average computation time for a valid result was 0.45 sec for the first experiment and 0.36 sec for the second experiment. Since the

solution of the preceding optimization are used as the initial trajectories for the solver, the solution converges faster if there is less change of the parameters between horizons. Currently, a limitation of the current design of the routine of the optimization module is the requirement of the preceding optimization being finished before the next optimization can be started. Consequently, the time between two horizons  $T_\Delta$  needs to be chosen sufficiently high to avoid additional delays from waiting for the preceding optimization to finish. Settings that affect the solution time include the number of collocation elements and the number of collocation points within the discretization of the optimization problem.

For verification purposes and to investigate the impact of the length of the horizon  $T_H$ , a test case has been constructed based on data obtained by a simulation of the model. Here, the exact parameter values that should result out of the identification are known. All three parameters have been varied during the simulation to evaluate the dynamical effects of the identification procedure. Using a sinus signal as input for  $P_{Ao}(t)$  and a constant signal for  $P_{CV}(t)$ , the resulting  $Q_{Ao}(t)$  was obtained. The so artificially created signals were used to emulate the sensor signals. The results of this test case are contained in Figure 8. As it can be seen, a smaller value for the horizon length  $T_H$  results in a faster response to changing parameters. On

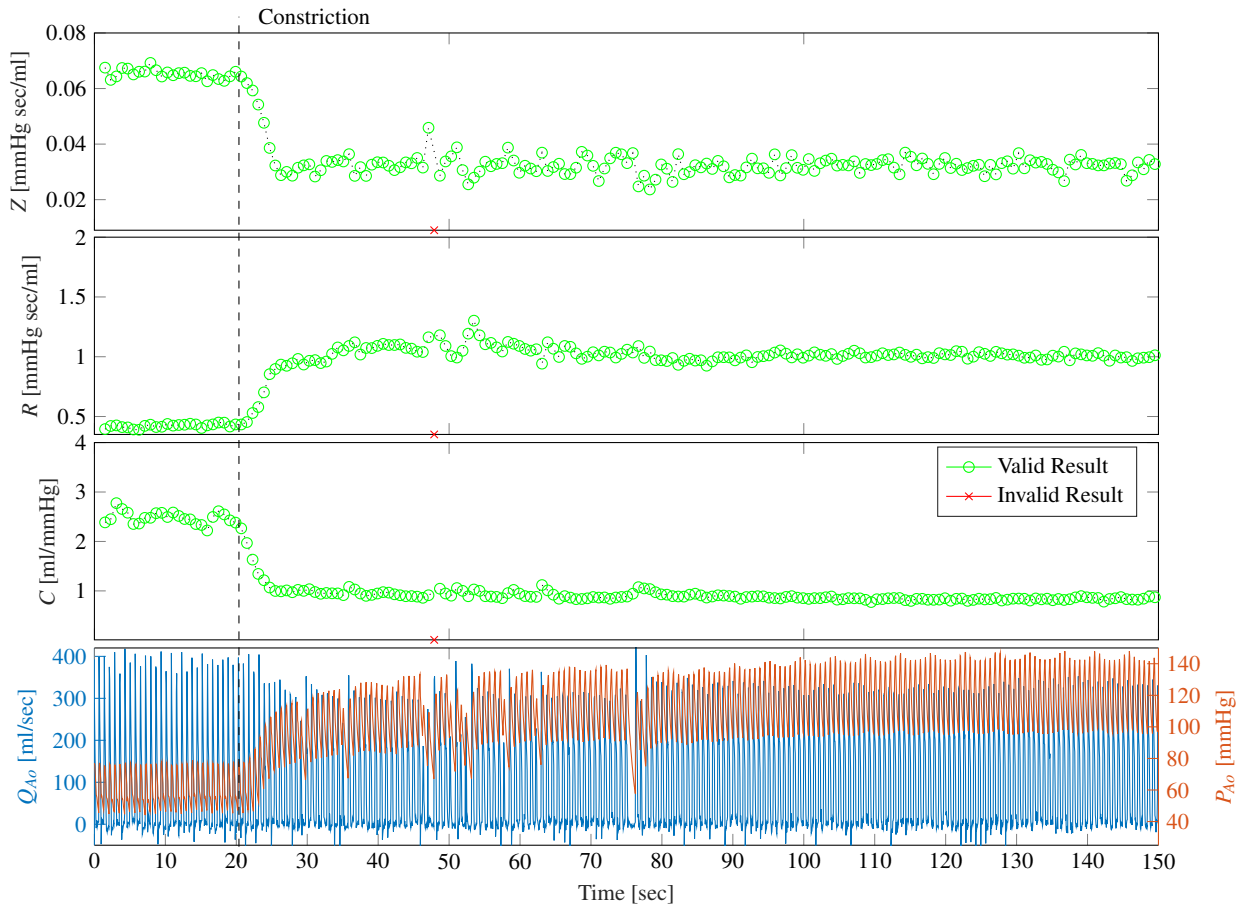


Figure 7. Long constriction at the descending position of the aorta.

the other hand, fluctuations are more damped for bigger values of  $T_H$ . Especially in the presence of noise, choosing a bigger value for  $T_H$  might be preferable. However, significant parameter changes within the horizon lead to invalid results (the outliers for  $T_H = 5$  sec in Figure 8).

In the future, it is planned to further evaluate the concept considering other aspects of the cardiovascular system. Besides, further improvements to the optimization module will be made to improve the robustness and the performance. This includes canceling optimizations that do not converge within a given time. Similarly, it will be considered to adapt the time between horizons  $T_\Delta$  depending on the necessary solution time. Another interesting idea that has come up is the synchronization of the horizons to heartbeats. Besides, improvements will be made to the setup work flow aiming at eliminating the need to adapt the Python code for new setups. Furthermore, we will be looking into changing the settings of within the optimization module through the *ModeliChart* interface without the need to stop and restart the optimization module.

Concluding, the concept that has been integrated into our lab infrastructure presents a valuable addition for our research on the cardiovascular system and has the potential to be used as a clinical tool in the future.

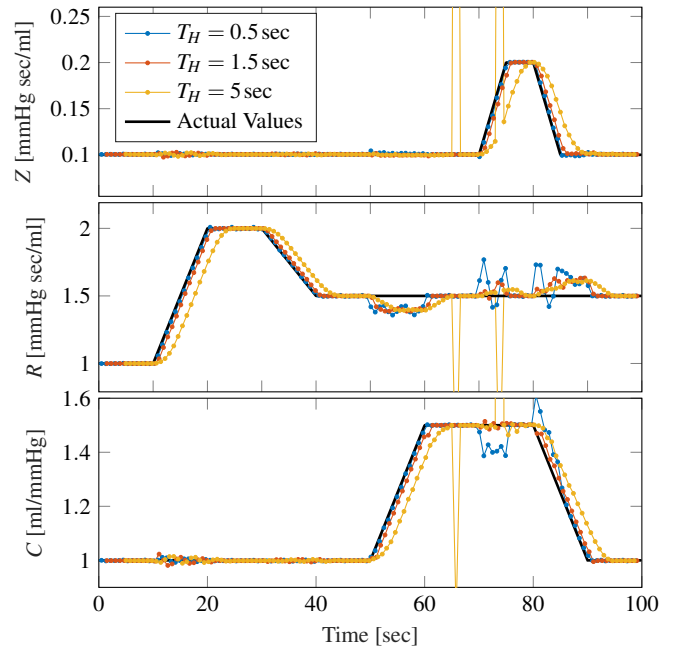


Figure 8. Verification trough identification of known parameter values for different horizon lengths  $T_H$ . The outliers in the  $T_H = 5$  sec test are invalid results.



## Acknowledgments

This work was supported by the German Research Foundation (DFG) within the Smart Life Support 2.0 PathoMod project (PAK 183-2).

## References

- J. Åkesson, K.-E. Årzn, M. Gäfvert, T. Bergdahl, and H. Tummescheit. Modeling and optimization with Optmica and JModelica.org—languages and tools for solving large-scale dynamic optimization problems. *Computers & Chemical Engineering*, 34(11):1737–1749, November 2010. doi:10.1016/j.compchemeng.2009.11.011.
- Joel Andersson, Johan Åkesson, Francesco Casellad, and Moritz Diehl. Integration of CasADi and JModelica.org. In *Proceedings from the 8th International Modelica Conference, Technical University, Dresden, Germany*, pages 218–231. Linköping University Electronic Press, June 2011. doi:10.3384/ecp11063218.
- A. Brunberg, J. Maschuw, R. Autschbach, and D. Abel. Object-oriented model library of the cardiovascular system including physiological control loops. In *IFMBE Proceedings*, pages 166–169. Springer Nature, 2009. doi:10.1007/978-3-642-03895-2\_48.
- John W. Clark, Robert Y. S. Ling, R. Srinivasan, J. S. Cole, and Roderick C. Pruet. A two-stage identification scheme for the determination of the parameters of a model of left heart and systemic circulation. *IEEE Transactions on Biomedical Engineering*, BME-27(1):20–29, January 1980. doi:10.1109/tbme.1980.326687.
- European Commission. Joint report on health care and long-term care systems & fiscal sustainability. *European Economy Institutional Papers*, (037), October 2016. doi:10.2765/680422.
- Jonas Gesenhues, Marc Hein, Moritz Habigt, Mare Mechelinck, Thivaharan Albin, and Dirk Abel. Nonlinear object-oriented modeling based optimal control of the heart: Performing precise preload manipulation maneuvers using a ventricular assist device. In *2016 European Control Conference (ECC)*. Institute of Electrical and Electronics Engineers (IEEE), June 2016. doi:10.1109/ecc.2016.7810603.
- Jonas Gesenhues, Marc Hein, Maïke Ketelhut, Moritz Habigt, Daniel Rüschen, Mare Mechelinck, Thivaharan Albin, Steffen Leonhardt, Thomas Schmitz-Rode, Rolf Rossaint, Rüdiger Autschbach, and Dirk Abel. Benefits of object-oriented models & ModeliChart: Modern tools and methods for the interdisciplinary research on smart biomedical technology. *Biomedical Engineering / Biomedizinische Technik*, 2017. doi:10.1515/bmt-2016-0074.
- Moritz Habigt, Maïke Ketelhut, Jonas Gesenhues, Frank Schrödel, Marc Hein, Mare Mechelinck, Thomas Schmitz-Rode, Dirk Abel, and Rolf Rossaint. Comparison of novel physiological load-adaptive control strategies for ventricular assist devices. *Biomedical Engineering / Biomedizinische Technik*, 2016. doi:10.1515/bmt-2016-0073.
- C.E. Hann, J.G. Chase, and G.M. Shaw. Integral-based identification of patient specific parameters for a minimal cardiac model. *Computer Methods and Programs in Biomedicine*, 81(2):181–192, February 2006. doi:10.1016/j.cmpb.2005.11.004.
- Maïke Ketelhut, Frank Schrödel, Sebastian Stemmler, Jesse Roseveare, Marc Hein, Jonas Gesenhues, Thivaharan Albin, and Dirk Abel. Iterative learning control of a left ventricular assist device. In *19th IFAC World Congress, Accepted for publication*. Toulouse, France, 2017.
- Ryo Kosaka, Yoshiyuki Sankai, Tomoaki Jikuya, Takashi Yamane, and Tatsuo Tsutsui. Online parameter identification of second-order systemic circulation model using the delta operator. *Artificial Organs*, 26(11):967–970, November 2002. doi:10.1046/j.1525-1594.2002.07112.x.
- Marek Mateják, Tomáš Kulhánek, Jan Šilar, Pavol Privitzer, Filip Ježek, and Jiří Kofránek. Physiobrary - modelica library for physiology. In *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*. Linköping University Electronic Press, March 2014. doi:10.3384/ecp14096499.
- Berno J.E. Misgeld, Daniel Rüschen, Sebastian Schwandner, Stefanie Heinke, Marian Walter, and Steffen Leonhardt. Robust decentralised control of a hydrodynamic human circulatory system simulator. *Biomedical Signal Processing and Control*, 20:35–44, July 2015. doi:10.1016/j.bspc.2015.04.004.
- Ajay Moza, Jonas Gesenhues, Dirk Abel, Rolf Rossaint, Thomas Schmitz-Rode, and Andreas Goetzenich. Patient specific parameter estimation for cardiovascular system models based on clinical measurements. *Biomedical Engineering / Biomedizinische Technik*, 2017. doi:10.1515/bmt-2016-0078.
- M Nichols, N Townsend, R Luengo-Fernandez, J Leal, A Gray, P Scarborough, and M Rayner. European cardiovascular disease statistics 2012. european heart network, brussels, european society of cardiology, sophia antipolis. 2012. *Cerebrovasc. Dis*, 25:457–507, 2012.
- Daniel Rüschen, Miriam Rimke, Jonas Gesenhues, Steffen Leonhardt, and Marian Walter. Online cardiac output estimation during transvalvular left ventricular assistance. *Computer Methods and Programs in Biomedicine*, August 2016. doi:10.1016/j.cmpb.2016.08.020.
- Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, April 2005. doi:10.1007/s10107-004-0559-y.
- Nico Westerhof, Jan-Willem Lankhaar, and BerendE. Westerhof. The arterial windkessel. *Medical & Biological Engineering & Computing*, 47(2):131–141, 2009. ISSN 0140-0118. doi:10.1007/s11517-008-0359-2.
- Yih-Choung Yu, J.R. Boston, M.A. Simaan, and J.F. Antaki. Estimation of systemic vascular bed parameters for artificial heart control. *IEEE Transactions on Automatic Control*, 43(6):765–778, June 1998. doi:10.1109/9.679017.