

Model Predictive Control of an Anesthesia Workstation Ventilation Unit [★]

Georg Männel* Marlin Siebert* Dennis Kleinewalter**
Christian Brendle** Philipp Rostalski*

* *Institute for Electrical Engineering in Medicine, University of Lübeck, Moislinger Allee 53-55, 23558 Lübeck, Germany (e-mail: ge.maennel, m.siebert, philipp.rostalski@uni-luebeck.de).*

** *Drägerwerk AG & Co. KGaA, Moislinger Allee 5355, 23558 Lübeck, Germany (e-mail:dennis.kleinewalter, christian.brendle@draeger.com).*

Abstract: Modern intensive care therapy as well as general anesthesia would not be possible, without respiratory support. Yet, unphysiological pressure levels and gas concentrations pose a serious risk to severely harm the patient. Advanced control schemes could improve the patient's safety and ensure the therapeutic success. Model predictive control (MPC) for instance allows to incorporate information about the patient at runtime through an internal model of the system, e.g. by using the lung compliance or airway resistance as model parameters. Furthermore, it can guarantee the satisfaction of constraints, which is useful, when considering physiological safety bounds. In this article we propose a two layered model-based control architecture for pressure controlled ventilation. The purpose of the lower layer is to approximately linearize the actuator dynamics, while the second layer implements a MPC controlling the pressure at the upper airways of the patient. The control architecture is implemented in an experimental setup, incorporating the ventilation unit of an anesthesia workstation. Initial results are presented, with the focus on the general feasibility of the chosen approach.

Keywords: Model Predictive Control, Respiratory Support Ventilation, Embedded System

1. INTRODUCTION

Respiratory support is a key element of modern medicine. By sustaining the ventilation of the patients lung, its task is to provide sufficient gas exchange during partial or complete failure of the patients respiratory system (Shelly and Nightingale, 1999). It is thus essential during general anesthesia to maintain the patients oxygenation and eliminate the produced carbon dioxide in case of the respiratory muscles are fully or partially relaxed (Shelly and Nightingale, 1999). Due to the applied positive pressure and commonly increased oxygen concentration in the inspired air, mechanical ventilation is non-physiological, with a high potential of severely damaging the lung of the patient (Shelly and Nightingale, 1999). Furthermore, it affects the cardiovascular system by elevating its workload, due to the increased counter pressure applied by the ventilation system (Shelly and Nightingale, 1999). Thus, the safety criteria on ventilation units are rigorous and highly regulated.

By using modularisation together with a model-based control approach the complexity of the resulting control system can be coped with, helping to ensure a safe operation of the medical ventilation unit. Männel et al. (2018) proposed a hierarchy of segregated control loops for the development of a respiratory support system alongside a model predictive controller for the carbon dioxide gas ex-

change, covering the higher layer of this hierarchy. In this article, we introduce a two level model-based control approach for controlling the pressure level of the ventilation unit of an anesthesia workstation, thus covering the lower layer control loops of the proposed hierarchy, Männel et al. (2018). These loops comprise of an actuator level controller abstracting the nonlinear actuator dynamics and a second level pressure controller. For this level, a model predictive control scheme is proposed to control the ventilation of the patients lung, whilst satisfying patient-safety constraints. In order to show the general feasibility of this model-based approach on a resource-constraint system, the resulting control architecture is implemented on two cortex M4 microcontrollers.

To the best of the authors knowledge Li and Haddad (2013) were the first to propose MPC for use in respiratory support. They presented simulation results following reference volume pattern to ventilate a multicompartment lung model also considering physiological constraints. Furthermore Scheel et al. (2017) proposed the use of MPC for the pressure regulation of the patient airways. The approach was proposed for a medical device maintain a continuous positive airway pressure (CPAP) commonly used in the therapy of obstructive sleep apnea with a patient breathing on its own. The controller objective was to counter act disturbances, such as the patient breathing effort and did not incorporate any constraints on the system. Our goal in contrast is to also support patients unable to breath spontaneously. In this setting the incorporation of physiological

* This work was partially funded by the Drägerwerk AG & Co. KGaA.

constraints is highly desirable. The proposed approach is illustrated both in simulation and with experimental results.

Notations: The weighted euclidean norm of vector $x \in \mathbb{R}^n$ is defined as $\|x\|_M = \sqrt{x^T M x}$, where $M \in \mathbb{R}^{n \times n}$ is a symmetric positive semi-definite matrix. For a sequence $\mathbf{u} \triangleq [u(0), u(1), \dots, u(N-1)]$, $u(i)$ denotes the i^{th} element of \mathbf{u} . A *polyhedron* is the intersection of a finite number of halfspaces $P = \{x \mid Ax \leq b\}$ and a *polytope* is a closed bounded polyhedron. The identity matrix of dimension n is denoted by I_n , while $\mathbf{0}_{n,p}$ denotes a zero matrix within the $\mathbb{R}^{n \times p}$.

2. SYSTEM DESCRIPTION

The ventilation unit is an important component of the anesthesia workstation. In order to illustrate the applicability of the proposed control approach the ventilation unit of an anesthesia workstation, a semi-closed rebreathing circuit is used as basis for our investigations. Semi-closed means that the exhaled air is partially reused after removing the exhaled carbon dioxide by means of an absorber. A schematic of the simplified pneumatic system is displayed in Figure 1. Depending on the clinical situation, the ventilation is accomplished via a mask or a tubus. Either one is attached to a y-piece from where two hoses are connecting the patient to the system, for inspiration and expiration respectively. During each breathing phase the gas flows through one of the hoses, which is ensured by a check valve marked as I for inspiratory and E for expiratory branch (see figure 1). The pressure at the y-piece, representing the pressure at the patient's airways, is controlled by two actuators. In the inspiratory branch a blower is located directly before the check valve. In the expiratory branch a valve is placed directly behind the check valve. Since this valve is essential for maintaining a positive end expiratory pressure (PEEP), it is called PEEP-Valve. During the inspiration the the blower increases the inspiratory pressure (p_{insp}) while the PEEP-valve is closed. Thus, air flows into the direction of the patient, filling its lung. The pressure is kept at the positive inspiratory pressure level (PIP) as depicted in figure 1.

Once the expiration phase starts the blower reduces the pressure p_{insp} and air is released from the lungs through the opening of the PEEP-valve. It is common to not release all air from the lung, but holding a positive pressure level, in order to prevent the alveoli from collapsing. The expiration is mainly driven by the restoring forces within the lung, thus the transient behavior during expiration can only be manipulated within tight bounds.

The exhaled air is collected in a reservoir bag, from which the blower can suck in air. This air is cleansed of CO_2 by an absorber, before it is redirected to the patient. The reuse of the exhaled air reduces the amount of required anesthetics and oxygen.

In order to compensate the oxygen uptake of the patient and to change the mixture of the breathing gas if required, fresh gas (FG) flows constantly into the breathing system. When a fast change of gas mixture is desired, it is possible to use a constant circular flow through the system. Furthermore, an adjustable pressure limitation (APL) is

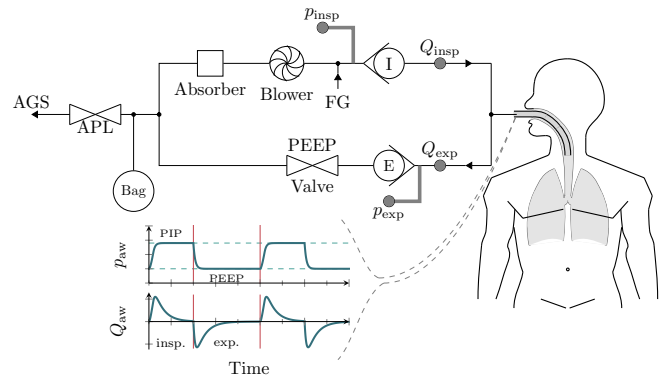


Fig. 1. Simplified schematic representation of the ventilation system and the typical flow and pressure waveforms at the patients upper airways during pressure controlled ventilation

required as safety measure for manually ventilating the patient, e.g. in case of a complete system failure.

Finally two valves, not shown in figure 1, are used to keep the pressure in the rebreathing branch close to atmospheric pressure. If the pressure is above atmospheric pressure, gas is released into the scavenging system (AGS) and otherwise gas is drawn from the atmosphere.

For the experimental setup used throughout this work, a modular research demonstrator of an anesthesia device is used. The demonstrator is based on the rebreathing circuit and actuators of a commercially available anesthesia workstation and illustrates a bus-modular design concept. The ventilation system is controlled by two microcontrollers, one Infineon XMC-4800 and one Infineon XMC-4300, both using the ARM Cortex-M4 architecture. Both controllers are exchanging data over an EtherCAT-bus with a sampling time of 1 ms. The bus master is implemented on a Raspberry pi 3B+ with an rt-patched raspian operating system. The XMC-4300 is used to read and pre-process the data from the sensors. The other microcontroller is connected to the systems actuators and provides a larger internal memory, thus the control architecture for the ventilator unit is implemented there.

3. SYSTEM MODEL

A model describing the system dynamics of the breathing system is derived using first principle methods. The model shall be used within the MPC, where it is beneficial to use a linear model with linear constraints to reduce the complexity of the optimization problem. Therefore, the rebreathing circuit of the ventilation system is neglected and the actuators are assumed to behave like linear first order systems. The differential equation for the inspiratory pressure p_{insp} , for the blower, and expiratory flow Q_{exp} , for the PEEP-valve, is given by:

$$\begin{aligned} \dot{p}_{\text{insp}} &= \tau_{\text{Blower}} p_{\text{insp}} + p_{\text{insp},r}, \\ \dot{Q}_{\text{exp}} &= \tau_{\text{PEEP}} Q_{\text{exp}} + Q_{\text{exp},r}, \end{aligned} \quad (1)$$

with $p_{\text{insp},r}$ and $Q_{\text{exp},r}$ being the respective input. The time constant of the actuator is given by τ_{Blower} and τ_{PEEP} respectively. The actuators are yet bound to constraints. On the one hand, the blower can only generate positive

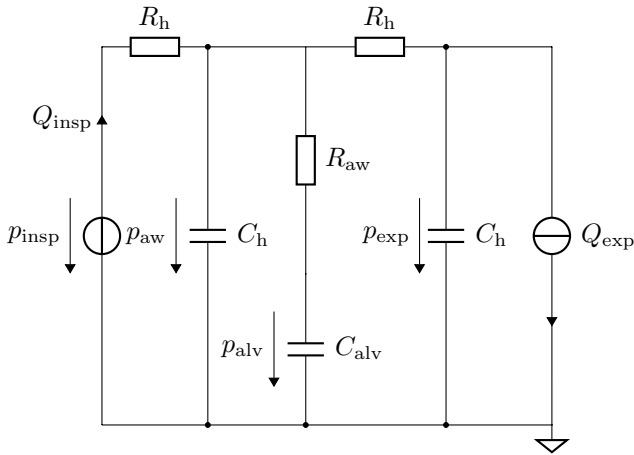


Fig. 2. Electrical analog of the ventilation system

pressure and has an upper limit. On the other hand, the maximal flow through the expiratory valve is dependent on the pressure p_{exp} at the valve. Under the assumption that the air flow is laminar, the following inequality holds:

$$Q_{exp} \leq G_{max} p_{exp},$$

where G_{max} is the conductance of the completely opened valve.

The ventilation hose poses a resistance (R_h) to the air flow and increases its volume, thus it behaves like a compliance (C_h). In combination the ventilation hose is modeled as a first order low-pass filter.

For the patient a single compartment model is used, where it is assumed that the patients airways behave as one rigid tube and the lung can be modeled as a single elastic compartment (Bates, 2009). Under the assumption of laminar air flow, the change of pressure within the alveoli p_{alv} can be expressed as

$$\dot{p}_{alv} = \frac{1}{R_{aw}C_{alv}}(p_{aw} - p_{alv}),$$

where p_{aw} is the pressure at the mouth piece or tubus of the patient. The parameter R_{aw} , C_{alv} describe the airway resistance and lung compliance respectively.

The electrical analog, where the volume flow and pressure are assumed to be an analog to current and voltage of the connected components is displayed in figure 2. Note that the pressure in the inspiratory hose, i.e. at the y-piece, is assumed as the airway pressure p_{aw} . The expiratory pressure is equivalent to the pressure in the expiratory hose. The behavior of the check valves and thus the directed flow within this model is considered by applying additional constraints to the system. In the expiratory branch it is sufficient to demand that $Q_{exp} \geq 0$, which is easily represented as an input constraint. For the inspiratory branch the constraint $Q_{insp} \geq 0$ implies that $p_{insp} \geq p_{aw}$. Nevertheless, the introduced constraints seemed more promising for the current setup, than the introduction of switching behavior. Given this system, the derived state space model has five states, two inputs and two outputs, namely:

$$x = \begin{bmatrix} p_{alv} \\ p_{aw} \\ p_{exp} \\ p_{insp} \\ Q_{exp} \end{bmatrix}, \quad u = \begin{bmatrix} p_{insp,r} \\ Q_{exp,r} \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} p_{exp} \\ Q_{insp} \end{bmatrix}.$$

4. CONTROL ARCHITECTURE

In order to control the pressure at the patient's airways we propose a hierarchical controller design with two layers and an additional therapy control unit (TCU). A block diagram of the proposed system is depicted in figure 3. The purpose of the therapy control unit in the given scenario is to generate the reference for the pressure at the y-piece, expecting the parameters of the pressure curve as input. In a more general setting the TCU would also contain all the different aspects relevant for the therapy. Thus, higher level controller, e.g. for the gas exchange (Männel et al. (2018)), would be abstracted within this block.

Similarly to the TCU the lower level controllers, K_{Blower} and K_{PEEP} , are used to simplify the actuator dynamics by compensating their nonlinearity and counteracting disturbances. These controllers are designed and empirically tuned such as their closed loop dynamics can be considered as approximately linear but with constraints, i.e. limits in their output. The blower is controlled such that it can be considered as a pure pressure source generating p_{insp} , while the expiratory valve controls the outflow Q_{exp} of the system. The subordinate controllers therefore provide a means for the assumptions of (1) to actually hold. Furthermore, the controller counter act disturbances from the rebreathing circuit, allowing to neglect it in the model of the MPC. Both controllers at this level run with a sampling time of 1 ms and are truly distributed, sharing no information, and furthermore having no knowledge about the patient.

The actual ventilation of the patient is achieved by the second control layer, which uses model based optimization to guaranty safe and reliable operation. The proposed approach contains two elements, on the given hierarchy level. For one the actual model predictive controller (MPC) calculating the future inputs and a Kalman filter (KF) estimating the current state of the system. Independently from the actual control architecture, the state estimate might be a useful addition for the system supervision, e.g. to identify abnormal or faulty behavior (Mehra and Peschon, 1971). The internal model of the filter and the controller is a zero-order hold discretized version of the continuous time model at a sampling rate of 10 ms.

In the authors opinion this approach holds several advantages over a single central controller. For one the MPC scheme provides a means to incorporate constraints into the system and guarantees that those are not violated. This can be used to directly introduce soft and hard physiological boundaries increasing the patients safety during therapy. Another major benefit comes from the internal model, which can be easily adapted to different patients and in future be extended, e.g. by incorporating spontaneous breathing. Furthermore, the control hierarchy enables the exchange of its components, e.g. an actuator, where only the respective sub-controller needs to be updated.

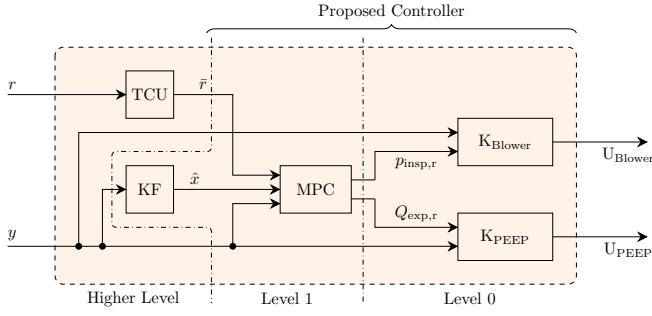


Fig. 3. Proposed control architecture for the ventilation unit

4.1 Model Predictive Controller

In the following a model predictive controller is designed for tracking piece-wise constant references as proposed by Limon et al. (2008). Therefore, consider a system with the linear discrete time system model

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (2)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$ represent the systems state, input and output at time instance $k \in \mathbb{N}$. The system dynamics are described by the matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, where the pair (A, B) is assumed to be controllable. The system output is determined by the system matrices $C \in \mathbb{R}^{p \times n}$ and $D \in \mathbb{R}^{p \times m}$. The system is subject to input and state constraints

$$(x, u) \in \mathcal{X} \times \mathcal{U}, \quad (3)$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$ are polyhedral sets, containing the origin as an interior point. Constraints on the system outputs are not considered explicitly, but rather transformed into state and input constraints using the affine mapping in (2). The solution to (2) at time i for an input sequence \mathbf{u} , is denoted $x(i) = \phi(i; x(0), \mathbf{u})$. The input sequence is considered to be an optimal input sequence \mathbf{u}^* , when it minimizes the quadratic cost function:

$$\begin{aligned} J(\mathbf{u}, \theta, x(0), y_r) &= \sum_{k=0}^{N-1} (\|x(k) - x_s\|_Q^2 + \|u(k) - u_s\|_R^2) \\ &\quad + \|x(N) - x_s\|_P^2 + \|y_s - y_r\|_T^2. \end{aligned} \quad (4)$$

The sum within the cost function accumulates for N consecutive time steps a weighted cost on the deviation between the predicted state and input to a steady state (x_s, u_s) . For the weighting matrices $Q = Q^T \succeq 0$, $R = R^T \succ 0$ and $T = T^T \succ 0$ hold. Furthermore, a cost term for the deviation between the state at the end of the horizon and the steady state is included approximating the cost of the infinite horizon optimal control problem, (Borrelli et al., 2017). Thus, the weighting matrix P is the solution to the discrete time Algebraic Riccati Equation

$$P = A^T P A - (A^T P B)(R + B^T P B)^{-1} (B^T P A) + Q.$$

The steady state is selected such that it minimizes weighted deviation between the steady state output and the reference y_r , while also being reachable without violating the system constraints at any time. Thus, each selected steady state (x_s, u_s) must satisfy the equation

$$\begin{bmatrix} A - I_n & B & \mathbf{0}_{n,p} \\ C & D & -I_p \end{bmatrix} \begin{bmatrix} x_s \\ u_s \\ y_s \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n,1} \\ \mathbf{0}_{p,1} \end{bmatrix}. \quad (5)$$

If the pair (A, B) is stabilizable, a non-trivial solution to (5) exist, (Muske and Rawlings, 1993) and can be parameterized by the parameter vector $\theta \in \mathbb{R}^o$, given in (4), and the affine mapping

$$\begin{bmatrix} x_s \\ u_s \end{bmatrix} = M_\theta \theta, \quad y_s = N_\theta \theta, \quad (6)$$

with the mapping matrices $M_\theta \in \mathbb{R}^{(m+n) \times o}$ and $N_\theta \in \mathbb{R}^{p \times o}$. As proposed by Limon et al. (2008) the parameterization is used to calculate an invariant set \mathcal{X}_f for tracking given a state feedback controller K , rendering the closed loop system $A + BK$ Hurwitz. The calculated set is used to impose a constraint onto the state at the end of the horizon, demanding it to be within the set \mathcal{X}_f .

Given the current state of the system x and a reference y_r to be tracked the ideal input sequence \mathbf{u}^* is determined by solving the optimization problem

$$\begin{aligned} \min_{u, \theta} & J(u, \theta, x(0), y_r) \\ \text{s.t.} & x(0) = x, \\ & (2), (3), (6), \\ & (x(N), \theta) \in \mathcal{X}_f, \end{aligned} \quad (7)$$

where \mathbf{u} and θ are the decision variables. The stated optimization problem yields a standard quadratic programming problem, which can be efficiently solved using specialized solvers. The system is controlled using the receding horizon strategy, applying the first element of \mathbf{u}^* to the system and solving (7) again for the measured state at the next time instant.

4.2 Implementation

The complete proposed control architecture is implemented on a single microcontroller, which is programmed in C, using code generated by Matlab Simulink. A non-linear model, which forms the foundation of the controller development, was deduced using first principle methods. Unknown system parameters were estimated from measurements by means of system identification. The interfaces of the model are selected to match the interfaces of the real system. Thus, the model outputs each represent an actual sensor within the real system with an identical discretization and sampling rate. The model inputs correspond to the electronic signals controlling the actuators.

The implementation of the TCU, the Kalman Filter and subordinate controllers is accomplished by using components, for which code can be directly generated. The implementation of the MPC is segregated into two parts, a solver and a design part. As a solver, qpOASES (Ferreau et al. (2014)) is selected, since it is open source and a library free translation to C-code already existed. The solver expects a parameterization in the standard quadratic program form

$$\begin{aligned} \min_z & \frac{1}{2} z^T H z + 2z^T F \\ \text{s.t.} & b_{lG} \leq Gz \leq b_{uG} \\ & b_l \leq z \leq b_u, \end{aligned} \quad (8)$$

with the optimization variable z , the Hessian $H = H^T \succeq 0$ and the gradient vector F . All constraints are formulated

as linear inequalities, either as bounds (b_l , b_u) on the decision variables or as bounds (b_{lG} , b_{uG}) on the affine mapping G of the decision variables. Therefore to interface the solver, the optimization problem in (7) has to be converted to the form of (8). This transformation can easily be accomplished in Matlab using YALMIP (Löfberg, 2004), allowing the problem to be directly implemented as stated in (7). The computation of the terminal set \mathcal{X}_f is done using the Multi-Parametric Toolbox (Herceg et al., 2013). The required matrices are extracted from internal YALMIP model and used to interface the solver directly within the Simulink model and later on in the C-code on the microcontroller. At run time only the matrix parts, depending on the initial state and the reference, e.g. the gradient vector, have to be updated between solving steps.

Due to limited resources, especially memory and computational power, of the microcontroller a further optimization of the implementation is necessary. The lack of memory is particularly problematic, due to the fact, that the embedded version of qpOASES does not support sparse matrices. Therefore the sequential representation of (7) is selected, e.g. eliminating the states as decision variables and only keeping the system inputs over the horizon. As Rawlings et al. (2018) points out, this reformulation drastically reduces the size of optimization problem. The size of the quadratic program is further reduced by eliminating the current state and the reference as decision variable, since these are fixed during each run. Nonetheless, both are incorporated in the quadratic program as parameters for the calculation of the gradient and the bounds of the affine constraints. Thus, only \mathbf{u} and θ remain as decision variables in (8).

5. SYSTEM PERFORMANCE

The international standard for basic safety and performance of critical care ventilators (ISO 80601-2-12:2011) specifies that for different test lungs of different compliance and resistance 30 consecutive breathing cycles are measured for different PEEP and PIP-levels as well as breathing frequencies and inspiration times. In order to evaluate the proposed control setup for the given system one of the test case defined by the standard is used. A passive test lung with a compliance of 20 ml/mbar and resistance of 20 mbar/l/s is attached to the y-piece of the breathing system and the demanded breath cycles, with a frequency of 8 min^{-1} and inspiration time of 3 s, are measured. The selected pressure levels for inspiration and expiration are 15 mbar and 5 mbar respectively.

In order to evaluate the control system on the embedded system measures describing the performance on the microcontroller are as much of interest as the measures of the closed loop performance in general. In particular the memory size of the binary file on the microcontroller as well as the run time of the solver for each solved optimization problem are of interest. Concerning the closed loop system performance only a maximum overshoot of 10% or 2 mbar, whichever is greater, is acceptable. This requirement stems from the documentation of the original anesthesia workstation. Furthermore an undershoot during the expiration, critical when it comes to the therapy, is to be avoided.

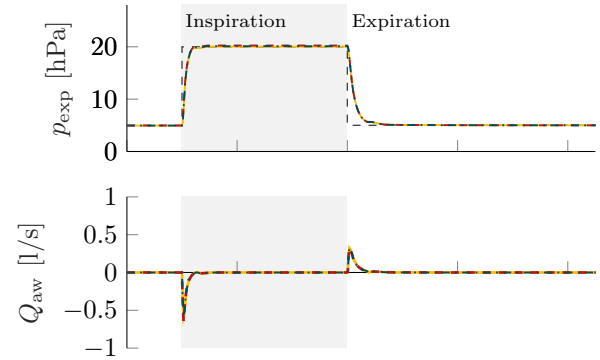


Fig. 4. Expiratory pressure (top) and patient flow (bottom) of the simulated breath cycle for different prediction horizon lengths ($N, Color, Type$) = $\{(2, red, dashed), (5, blue, dashdotted), (10, yellow, solid)\}$.

5.1 Numerical Results

The control approach is evaluated first in simulation using a nonlinear system model. The controller is tested for the horizon lengths of $N = 2, 5, 10$, from which the most suitable horizon will be implemented on the embedded system. Since no large deviation between breathing cycles is to be expected in simulation, only a single breath is considered instead of 30 as required by the norm.

In figure 4 the result of the simulations are depicted. The upper graph shows the pressure at the PEEP-valve, as surrogate to the airway pressure, and the lower graph shows the flow towards the patient. The closed loop performance of all three controller is quite similar. Actually only during the inspiration a deviation between the controllers can be quantified. The maximum overshoot for the controller with $N = 2$ (red, dashed) is 0.250 mbar and thus more than double the maximum overshoot of 0.114 mbar for the controller with $N = 5$ (blue, dash-dotted). The overshoot can be reduced even further to 0.027 mbar with the horizon length of $N = 10$. The controller with the shortest prediction horizon also shows a slight tracking error, which is not present for the other controllers. During the expiration all three controllers display a similar performance, which makes sense considering, that the dynamics of the expiration is mostly driven by the passive relaxation, thus the expiration can only be manipulated to a limited extent.

Regarding the amount of required memory for the optimization problem in relation to the horizon length can be expressed by a quadratic function. When operating with double precision floating point numbers the minimal required memory for given optimization problems is 11.120 kB for $N = 2$, 19.496 kB for $N = 5$ and 37.936 kB for $N = 10$. Since the solver keeps more than one copy of these matrices in the memory at any time, the real memory requirement is significantly larger.

5.2 Experimental Results

All three controllers satisfy the performance requirements given the maximum overshoot during inspiration and no undershoot during expiration in simulation. For the implementation on the embedded system the controller

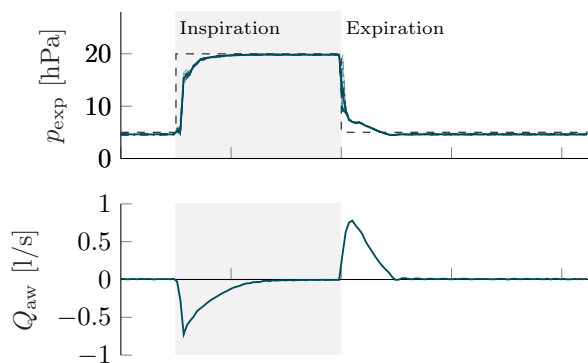


Fig. 5. Experimental results for the controller with prediction horizon $N = 5$ for 30 consecutive breathing cycles overlaid in one (shaded) and the mean value (solid) for each point on the respiratory cycle. The top graph shows the expiratory pressure and the bottom graph the patient flow.

with $N = 5$ was selected, since the controller with $N = 2$ shows a slight tracking error and the performance gain for $N = 10$ does not outweigh the increased complexity and memory requirement.

Figure 5 show the pressure measured at the PEEP-valve and the flow into the patient for all 30 respiratory cycles (thin dash dotted lines) overlaid in a single breath for each horizon length. Furthermore, it displays the mean value (solid line) of all breaths at any point in the breath cycle. The deviation between the single breath cycles is rather small and really noticeable in the onset of the inspiration and expiration.

The proposed controller is capable of following the given pressure reference trajectory (dashed dark line) reasonably well. The inspiratory rise time of the pressure is significantly slower compared to the simulation, but the maximal overshoot over all breath cycles is with 0.069 mbar well below the bound set by the requirement. During the expiration on the other hand, the undershoot is larger and a constant offset is maintained. Furthermore, the controller does not close the PEEP-valve entirely, which results in a small but undesirable constant circular flow, which is not depicted. Therefore, the pressure at the y-piece is actually slightly higher than measured, due to the pressure drop across the ventilation hose.

This shows the general applicability of the proposed control approach for the given system setup. When analyzing the time required for solving the optimization problem, it was noticed that only 87% of the time the problem could be solved for optimality within the sampling time of 10 ms. The origin of this issue requires further investigations and might increase the over all performance.

6. CONCLUSION

In this paper we demonstrated the applicability of hierarchical control architecture for pressure controlled ventilation on the ventilation unit of a bus-modular research demonstrator. The introduction of subordinate controllers allowed us to abstract the nonlinear actuators for the second level model predictive controller. The model predictive controller incorporated physiological and technical

constraints, holding the potential to increase the patients safety and incorporate therapeutic information through the model. The first results are promising, showing that it is possible to follow a desired breathing pattern while satisfying the formulated performance measures in simulation and on the embedded system.

REFERENCES

- Bates, J.H.T. (2009). *Lung Mechanics*. Cambridge University Press. doi:10.1017/cbo9780511627156.
- Borrelli, F., Bemporad, A., and Morari, M. (2017). *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press. doi:10.1017/9781139061759.
- Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., and Diehl, M. (2014). qpOASES: a parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4), 327–363. doi: 10.1007/s12532-014-0071-1.
- Herceg, M., Kvasnica, M., Jones, C.N., and Morari, M. (2013). Multi-parametric toolbox 3.0. In *2013 European Control Conference (ECC)*. IEEE. doi: 10.23919/ecc.2013.6669862.
- ISO 80601-2-12:2011 (2011). Medical electrical equipment - part 2-12: Particular requirements for basic safety and essential performance of critical care ventilators. Standard, Geneva, CH.
- Li, H. and Haddad, W.M. (2013). Model predictive control for a multicompartment respiratory system. *IEEE Transactions on Control Systems Technology*, 21(5), 1988–1995. doi:10.1109/tcst.2012.2210956.
- Limon, D., Alvarado, I., Alamo, T., and Camacho, E. (2008). MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9), 2382–2387. doi:10.1016/j.automatica.2008.01.023.
- Löfberg, J. (2004). YALMIP : a toolbox for modeling and optimization in MATLAB. In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*. IEEE. doi: 10.1109/cacsd.2004.1393890.
- Männel, G., Hoffmann, C., and Rostalski, P. (2018). A robust model predictive control approach to intelligent respiratory support. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. doi:10.1109/ccta.2018.8511363.
- Mehra, R. and Peschon, J. (1971). An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7(5), 637–640. doi:10.1016/0005-1098(71)90028-8.
- Muske, K.R. and Rawlings, J.B. (1993). Model predictive control with linear models. *AIChE Journal*, 39(2), 262–287. doi:10.1002/aic.690390208.
- Rawlings, J.B., Mayne, D.Q., and Diehl, M.M. (2018). *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, LLC.
- Scheel, M., Berndt, A., and Simanski, O. (2017). Model predictive control approach for a CPAP-device. *Current Directions in Biomedical Engineering*, 3(2). doi: 10.1515/cdbme-2017-0065.
- Shelly, M.P. and Nightingale, P. (1999). ABC of intensive care: Respiratory support. *Brit. Med. J.*, 318(7199), 1674–1677. doi:10.1136/bmj.318.7199.1674.