ABSTRACT

We have designed and implemented a generic virtual mechanical ventilator model into the open-source Pulse Physiology Engine for real-time medical simulation. The universal data model is uniquely designed to apply all modes of ventilation and allow for modification of the fluid mechanics circuit parameters. The ventilator methodology provides a connection to the existing Pulse respiratory system for spontaneous breathing and gas/aerosol substance transport. The existing Pulse Explorer application was extended to include a new ventilator monitor screen with variable modes and settings and a dynamic output display. Proper functionality was validated by simulating the same patient pathophysiology and ventilator settings virtually in Pulse as a physical lung simulator and ventilator setup.

Keywords: mechanical ventilator, physiology, open-source, pulse engine, respiratory

1 INTRODUCTION

1.1 The Pulse Physiology Engine

The Pulse Physiology Engine (Kitware 2022a) is an open-source, well-validated, comprehensive human physiology software toolkit that has been leveraged for many diverse medical simulation applications (Bray et al. 2019, Kitware 2022b). The engine is comprised of numerical models representing the different physiological systems of the body, feedback mechanisms and interactions between the systems, pharmacokinetic/pharmacodynamic (PK/PD) behavior (Clipp et al. 2016), and medical equipment. The major systems are modeled using zero-dimensional (0-D) lumped-parameter circuit analogs with homeostatic feedback. The differential equations contained in each system are calculated through transient analysis with a shared dynamic time-step of 20 milliseconds, while maintaining the ability to simulate faster than real-time.
The platform includes a common data model (CDM) for standard model and data definitions, a software interface for engine control, robust physics-based circuit and transport solvers, and a verification and validation (V&V) suite. The Pulse Software framework is implemented in C++ and provides bindings to various popular programming languages including C, C#, Java, and Python. Depending on the target application platform, Pulse can easily be integrated in both hardware and software systems using any of these languages.

1.2 Mechanical Ventilator Model Needs

A robust Pulse mechanical ventilator model is needed for several different use cases, including clinical investigations (Webb et al. 2020), training simulators (Rodriquez Jr. et al. 2019, Couperus et al. 2020), and hardware-in-the-loop applications (Gessa et al. 2018). This new ventilator model required a generic representation of a positive-pressure ventilation device with functionality for inhaled gas/agent administration. It needed to integrate with the existing Pulse respiratory fluid mechanics circuit and transport graph (Webb, Bray, and Clipp 2020, Webb 2022) using the same physics-based fidelity previously implemented with legacy equipment models. Therefore, a mechanistic modeling approach was employed to allow all existing Pulse functionality for patient-specific physiologies and pathophysiologies to be simulated with the addition of artificial ventilation. Feedback between the respiratory system and the mechanical ventilator was required to trigger assisted modes from patient spontaneous breaths, while still allowing for equipment initiated modes of ventilation. The new Pulse mechanical ventilator methodology also needed to be modular to allow the simulation of all conventional and adaptive ventilation modes (Lei 2017), as well as various equipment brands and models with disparate definitions for settings, sensors, and interfaces. Extensibility was required to interface with real-world cyber-physical systems, including prototype equipment for in silico trials.

While other ventilator virtual software and computational models exist (Hannon et al. 2022, Lino et al. 2016), none are suitable for direct integration into Pulse. Existing software is often hardcoded with other physiology models and not modular enough for extraction, not realistic/high-fidelity enough for integration with Pulse’s lumped parameter approach, too focused on specific modes and pathophysiological states, and/or is closed-source without a permissive license. Therefore, a new approach was required.

1.3 Taxonomy for Mechanical Ventilation

The terminology of ventilation modes is not globally standardized. Ventilator manufacturers and researchers name new modes even for trivial changes or innovations (Lei 2017). The consequence is a chaotic pool of mode terminology, where the number of modes of ventilation has grown exponentially in the last three decades. Existing respiratory care equipment includes hundreds of unique names of modes on more than 30 different ventilators. To meet the needs of a Pulse mechanical ventilator model, all existing modes of ventilation were generalized to an accurate and manageable set of configuration parameters. This requires an understanding of the base taxonomy of mode definitions used by respiratory care professionals. Chatburn, El-Khatib, and Mireles-Cabodevila (2014) classified modes of ventilation through 10 fundamental constructs of ventilator technology in order to define the three basic components: (1) ventilator breath control variable, (2) the breath sequence, and (3) the targeting scheme. We have leveraged this theoretical basis of taxonomy and standard ontologies (Carlo, Ambalavanan, and Chatburn 2012) for our definitions of ventilator settings that are exposed to Pulse users and applications.
2 METHODS

2.1 Data Model

The Pulse CDM (Bray 2022a) and application programming interface (API) (Bray 2022b) are designed to support computational physiology modeling by providing a standardized data interchange between physiology models and outside software. The CDM further supports distributed integration through the use of its lightweight action data structures that can easily be added to any network-based protocol. We have designed the ventilator to integrate with the existing architectures using these Pulse software framework features.

The goal of the ventilator data model is to distill all possible mode taxonomies into a minimal, generalized set of configuration parameters that can be parsed by the internal engine methodology. To achieve this, the settings do not include mode definitions specifically, but are instead a logical layer that defines the lower level parameters necessary for breath phase definitions. Much of the mechanical ventilator data model is focused on the calculation of the instantaneous pressure/flow source required to drive the circuit (see 2-4 below). These temporal functions are an important part of the transient fluid mechanics analysis used in Pulse simulations. Each time-step the ventilator model must update circuit parameters to their proper value based on the current phase in the breath cycle and configuration settings. Other configuration parameters define the fluid mechanics circuit elements (see 1 and 6-9 below) and values for the substance transport graph (see 5 below).

The Mechanical Ventilator configuration parameters were defined to allow for setting any ventilation modes - all control variable types and all breath sequences. While these definitions are intended to be all-encompassing, the data model has been implemented to allow for the addition of new features in the future. Logical checks are included to ensure minimal required settings are defined and conflicting values are handled. The hierarchical configuration definitions are:

1. Connection: Connection type to the respiratory system (Off, Mask, Tube)
2. Inspiration Phase
   (a) Cycle: Transition to pause/expiration
      i. Machine Cycle: Control cycles
         A. Time: Total length of inspiration phase to trigger expiration phase
      ii. Patient Cycle: Assisted cycles
         A. Pressure: Ventilator sensor pressure value to trigger expiration phase
         B. Flow: Ventilator sensor flow value to trigger expiration phase
         C. Volume: Ventilator sensor volume change (tidal volume) to trigger expiration phase
         D. Respiratory Model: Trigger expiration phase from respiratory model event
   (b) Waveform: Pattern of driver function (square, exponential, ramp, sinusoidal, sigmoidal)
   (c) Waveform Period: Time to reach maximum driver value
   (d) Target: Driver value set-point
      i. Pressure (PIP): Extrinsic pressure above atmosphere at the end of inhalation
      ii. Flow: Air flow at end of inhalation
   (e) Limit: Cutoff/Maximum
      i. Pressure: Ventilator sensor pressure cutoff/maximum
      ii. Flow: Ventilator sensor flow cutoff/maximum
      iii. Volume: Ventilator sensor total lung volume cutoff/maximum
   (f) Pause: Time of plateau (i.e., constant driver pressure) between inspiration and expiration
3. Expiration Phase
   (a) Trigger: Transition to inspiration
      i. Machine Trigger: Control triggers
A. Time: Total length of expiration phase to trigger inspiration phase
   ii. Patient Trigger: Assisted triggers
      A. Pressure: Ventilator sensor pressure value to trigger inspiration phase
      B. Volume: Ventilator sensor volume change value to trigger inspiration phase
      C. Respiratory Model: Trigger inspiration phase from respiratory model event
   (b) Waveform: Pattern of driver function (square, exponential, ramp, sinusoidal, sigmoidal)
   (c) Waveform Period: Time to reach minimum driver value
   (d) Baseline: Value to set/achieve
      i. Pressure (PEEP): Extrinsic pressure above atmosphere at the end of exhalation
      ii. Volume (FRC): Total lung volume at the end of exhalation
4. Driver Damping Parameter: Fractional change parameter to prevent abrupt changes and discontinuities (i.e., smoother driver curve)
5. Substances
   (a) Fraction of inspired gas: FiO2 and other gases fractions
   (b) Concentration of inspired aerosol: albuterol, etc.
6. Circuit Resistances
   (a) Inspiration tube resistance: Total resistance of inspiratory limb tubing
   (b) Inspiration valve resistance: Total resistance of inspiratory valves
   (c) Expiration tube resistance: Total resistance of expiratory limb tubing
   (d) Expiration valve resistance: Total resistance of expiratory valves
7. Circuit Volumes
   (a) Inspiratory limb volume: Total volume of inspiratory limb tubing
   (b) Inspiratory valve volume: Total volume of inspiratory valves
   (c) Expiratory tube volume: Total volume of expiratory limb tubing
   (d) Expiratory valve volume: Total volume of expiratory valves
   (e) Y-piece volume: Total volume of the y-piece
   (f) Connection volume: Total volume of the mask/endotracheal tube
8. Circuit Compliance: Total compliance of the entire ventilator circuit
9. Relief Valve Pressure: Maximum relative pressure allowed

Two user actions are also included in the virtual ventilator model: 1) a hold action that maintains the current driver pressure and can be set to inspiratory, expiratory, or instantaneous for application timing, and 2) a leak action that allows air to escape the connection with the respiratory system using a severity of leak scale.

The Pulse API allows compartment fluid (i.e., pressures, flows, volumes) and substance data (i.e., substance volumes and fractions) to be requested each time-step from anywhere in the ventilator circuit. However, there is a need to define further system data that may be more typical for display on monitors or requires further processing. Some of these outputs are instantaneous values that are updated each time-step (e.g., Airway Pressure), while others are calculated at transitions and remain unchanged until the next breath cycle (e.g., End Tidal Carbon Dioxide). Volume data is calculated by integrating the local flow over breath phases. Each phase is provided by the Breath State parameter that specifies patient or equipment initiated.

The following is a list of the mechanical ventilator system data (sampled in the ventilator circuit rather than the patient systems) and events that were added to the Pulse API:

- Airway Pressure: Instantaneous pressure applied during positive-pressure mechanical ventilation
- Breath State: Current breath phase type - No Breath, Patient Inhale, Patient Pause, Patient Exhale, Equipment Inhale, Equipment Pause, Equipment Exhale, Expiratory Hold, or Inspiratory Hold
- Dynamic Pulmonary Compliance: The pulmonary compliance during periods of gas flow
• End Tidal Carbon Dioxide Fraction: The fraction of carbon dioxide at the end of each cycle
• End Tidal Carbon Dioxide Pressure: The pressure of carbon dioxide at the end of each cycle
• End Tidal Oxygen Fraction: The fraction of oxygen at the end of each cycle
• End Tidal Oxygen Pressure: The pressure of oxygen at the end of each cycle
• Expiratory Flow: Instantaneous airflow out of the lungs (negative value when inhaling)
• Expiratory Tidal Volume: The volume of air moved out of the lungs during expiration
• Inspiratory:Expiratory Ratio: The ratio of the length of time of inspiration to the length of time of expiration
• Inspiratory Flow: The instantaneous airflow into the lungs (negative value when exhaling)
• Inspiratory Tidal Volume: The volume of air moved into the lungs during inspiration
• Intrinsic Positive End Expired Pressure: The pressure above ambient/atmospheric at the end of expiration - also known as auto-PEEP, air trapping, or dynamic hyperinflation
• Leak Fraction: The fraction of gas volume lost during the ventilation cycle determined from the air volume in and out difference
• Mean Airway Pressure: Mean pressure applied during positive-pressure mechanical ventilation
• Peak Inspiratory Pressure: The highest pressure applied to the lungs above ambient/atmospheric pressure during inhalation
• Plateau Pressure: The pressure applied by the ventilator measured at end-inspiration with an inspiratory hold maneuver
• Positive End Expiratory Pressure: Extrinsic pressure above atmosphere at the end of exhalation
• Pulmonary Resistance: The total resistance to airflow through the lungs
• Respiration Rate: The frequency of the respiratory cycle
• Static Pulmonary Compliance: The pulmonary compliance during periods without gas flow
• Tidal Volume: The total volume of air moved during respiration
• Total Lung Volume: The total volume of air within the lungs at any given time
• Total Pulmonary Ventilation: The amount of air that enters the lungs as a function of time - can be used for minute ventilation
• Start of Inhale Event: Starting inspiration phase
• Start of Exhale Event: Starting expiration phase
• Relief Valve Active Event: Relief valve is allowing air to escape to maintain the pressure setting

The mechanical ventilator model follows the same logic loop as all Pulse systems, as shown in Figure 1. The state at every time-step is determined through a three-step process: 1) a preprocess step determines the circuit element values based on feedback mechanisms and engine settings/actions, 2) a process step uses the generic circuit calculator and substance transporter to compute the entire state of the circuit and fill in all pertinent values, and 3) a postprocess step is used to advance time.

2.2 Methodology

The mechanical ventilator model consists of a pressure/flow source with tubes and valves for inspiration and expiration. The unidirectional valves are ideal and do not allow any backflow. Figure 2 shows the mechanical ventilator circuit. The compartments and transport graph mirrors the circuit. Substance values are set on the ventilator node/compartment, assuming infinite volume. The ventilator interacts with the existing respiratory circuit through a direct connection that allows air (gases and aerosols) to flow freely. A path connects the airway node of the respiratory system to the connection (mask or tube) node of the ventilator. When the machine is turned on, both individually defined circuits are combined into a single, closed-loop circuit that is solved as a single linear algebra matrix.
Figure 1: The mechanical ventilator software flow chart. Each loop represents a simulation time-step. Actions and configuration are applied at any time during the simulation when externally set. System data, compartment data, and events specific to the ventilator are exposed through the API.

Figure 2: The Pulse 0-D mechanical ventilator fluid circuit diagram with an example pressure source waveform. Node objects hold volumes and pressures, while paths hold element definitions and flows. The circuit employs a variable driver source (either pressure or flow), resistances, valves, and a compliance. The example driver waveform shown has a ramp waveform function for inspiration and square for expiration. The bracketed numbers and letters correspond to the enumerations previously listed in Section 2.1.
The software logic we implemented is meant to mimic the control software of real-world ventilators. Feedback is read from locations within the ventilator circuit as if they are from actual sensors. The data model defined in Section 2.1 was used to generate a data flow that is distilled down to simple conditionals, as shown in Figure 3. The model modifies the ventilator circuit elements each time-step based on the configuration settings and actions and the Pulse circuit solver and transporter handle the fluid mechanics and substance computations.

Figure 3: We implemented a phased breath logic in the Pulse mechanical ventilator methodology. The bracketed numbers and letters correspond to the enumerations previously listed in Section 2.1. Blue boxes signify transitions between the two phases that make up one full breath, white diamonds are conditionals, green boxes are the pressure/flow driver updates that occur each time-step, and grey boxes advance the simulation time to the next time-step. The ventilator can be stopped at any time-step during simulation.

The Pulse mechanical ventilator data model and methodology allows for simulating any mode of ventilation. However, external translations are required to map mode-specific settings to our generic virtual ventilator configuration parameters. Figure 4 shows examples of how to apply five common ventilator modes: pressure control - continuous mandatory ventilation (PC-CMV), volume control - continuous mandatory ventilation (VC-CMV), pressure control - assist control (PC-AC), volume control - assist control (VC-AC) ventilation, and continuous positive airway pressure (CPAP). These modes are included as actions in the API that will handle the mapping for users for more intuitive integration of Pulse into applications that require mechanical ventilation. Inspiratory Period ($T_i$) and Respiration Rate ($RR$) mode settings can be translated to the Inspiration Machine Trigger Time ($Te$) using $Te = \frac{1}{RR} - Ti$, and The Positive End Expired Pressure ($PEEP$) and Delta Pressure Support ($\Delta P_{supp}$) mode settings can be translated to the Peak Inspiratory Pressure ($PIP$) using $PIP = \Delta P_{supp} - PEEP$. 
Figure 4: Example mapping of common ventilator modes on the left to Pulse model parameters on the right. Black lines represent direct value application, while red lines represent relationships that require further calculations to translate. The bracketed numbers and letters correspond to the enumerations previously listed in Section 2.1.

3 RESULTS

3.1 Virtual Simulation

The Pulse Physiology Explorer (Bray 2022c) is an open-source application built on Qt. It provides visualization and the capability to intuitively interact with and view data generated by Pulse. We added a new page to the Explorer for simulating a ventilator monitor using the new mechanical ventilator API additions. Figure 5 shows screen shots from the Explorer with a new ventilator monitor page. We were able to leverage the Explorer to confirm proper functionality of the mechanical ventilator model and to ensure successful integration with all other Pulse physiology models.

3.2 Breathing Simulator Validation

To validate functionality of our virtual ventilator model, we compared Pulse simulation results with physical ventilator values and waveforms with the setup shown in Figure 6. The RespiSim software allowed us to set the ASL 5000 equations of motion patient respiratory parameters (i.e., resistances, compliances, and spontaneous breath waveforms) (IngMar Medical 2020) to the same values as the Pulse respiratory system. The same ventilator settings were also applied to the physical Dräger ventilator and the Pulse virtual ventilator model. We were able to create scenarios of spontaneously breathing patients with healthy lungs, restrictive disease, and obstructive disease in combination with PC-AC, VC-AC, and CPAP modes, for nine total validation use cases. Figure 7 shows some sample results.
Figure 5: The Pulse Explorer application is used to showcase and test the new mechanical ventilator model integrated with the existing whole-body computational physiology engine. The five modes shown in Figure 4 are exposed to the user with the ability to dynamically change settings. The left panel simulates a vitals monitor, the middle panel shows the new settings and outputs as a ventilator monitor, and the right panel is a popup for the volume-pressure and flow-volume loops.

Figure 6: The physical breathing simulator setup for virtual mechanical ventilator model validation is shown on the left. The ASL 5000™ lung simulator (Model 31 00 150, SN:2226) was connected with a network cable to a laptop running RespiSim® software (version 4.0.10520). A Dräger Evita® Infinity® V500 ventilator (SN:ASBD-0072) was connected with tubing to the ASL 5000. The patient respiratory physiological parameters and ventilator settings used for both the breathing simulator setup and the Pulse simulation are shown in the tables on the right.

The waveforms created by the virtual ventilator match well with the displayed physical ventilator monitor values in Figure 7. The important temporal features and patterns used by Respiratory Therapists to manage patients (Dexter and Clark 2020) are present. However, the data shows that an extended inspiration time by the physical ventilator translates to higher TVs than come from our ideal, deterministic circuit model. Discrepancies with the TV outputs for the PC-AC and CPAP modes can be attributed to imperfections with real-world sampling, lags from sensor triggering, and non-instantaneous driver transitions. This follows
expectations from previous lung model bench tests performed by l’Her, Roy, and Marjanovic (2014) that showed major differences between mechanical ventilator devices in both general characteristics and technical reliability. Most notable was large variability in TV delivery.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Healthy</th>
<th>Restrictive</th>
<th>Obstructive</th>
</tr>
</thead>
<tbody>
<tr>
<td>RR (bpm)</td>
<td>15</td>
<td>25</td>
<td>18</td>
</tr>
<tr>
<td>Pmean (cmH2O)</td>
<td>8.2</td>
<td>14</td>
<td>7.7</td>
</tr>
<tr>
<td>TV (mL)</td>
<td>918</td>
<td>569</td>
<td>518</td>
</tr>
</tbody>
</table>

4 DISCUSSION

The new Pulse mechanical ventilator methodology satisfies all of the requirements we set out to achieve, and will continue to be leveraged for many different users and use cases. Our validation analysis showed good

Figure 7: Results of the validation analysis from the setup shown in Figure 6. Each of the nine runs has a ground truth photo of the physical ventilator monitor screen, waveform results from the Pulse virtual ventilator, and a table comparing outputs between the two. The top waveform is the airway pressure, the middle is the inspiratory flow, and the bottom is the lung volume. \( RR \) is the respiration rate, \( P_{\text{mean}} \) is the mean airway pressure, and \( TV \) is the tidal volume.
agreement between the Pulse virtual ventilator and a real-world ventilator. This mechanical ventilator model can be used in a standalone Pulse application or integrated with simulators, sensor interfaces, and models of all fidelities. The data model has been carefully defined and implemented to allow for future expansion, including the implementation of patient-specific physiological and pathophysiological models.

ACKNOWLEDGMENTS

This research was partially supported by the National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health under award number R01EB031808. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

REFERENCES


**AUTHOR BIOGRAPHIES**

JEFFREY B. WEBB is a Technical Leader at Kitware, Inc. He holds a MS in Mechanical Engineering from Rochester Institute of Technology. His research interests include open-source computational modeling of physiological systems and their integration with cyber-physical systems. His email address is jeff.webb@kitware.com.

AARON BRAY is a Principal Engineer at Kitware, Inc. He holds a BS in Computer Science from Auburn University. His interests include software architecture for development of modeling and simulation codes. His email is aaron.bray@kitware.com.

JUSTINA GERARD is a Product Manager at IngMar Medical, LLC. She holds a MBA from California University of Pennsylvania. She is also Registered Respiratory Therapist. Her interests include product design for medical simulation. Her email is jgerard@ingmarmed.com.

STEFAN FREMBGEN is the CEO of IngMar Medical, LLC. He holds a PhD in Safety Engineering from the Bergische Universität Wuppertal, Germany. His interests include the production of test lungs and breathing simulators. His email is sfrembgen@ingmarmed.com.

HARALD SCHEIRICH is a Staff R&D Engineer at Kitware, Inc. He holds a Dipl. Ing. (MS Equivalent) in Computer Science from the Technical University, Vienna Austria. His interests are the design and development of surgical simulations and systems architecture. His email address is harald.scheirich@kitware.com.

JOSEPH VANPELT is a Senior R&D Engineer at Kitware, Inc. He holds a BS in Computer Science: Software Engineering from Herzing University. His interests include the design, development, and integration with cyber-physical systems. His email address is joseph.vanpelt@kitware.com.

RACHEL B. CLIPP, PHD is a Technical Leader at Kitware, Inc. She holds a MS and PhD in Biomedical Engineering from North Carolina State University and the University of North Caroline - Chapel Hill. Her research interests are medical modeling and simulation, including computational fluid dynamics, computational physiology, and surgical simulation. Her email address is rachel.clipp@kitware.com.