

Trabajo de Fin de Máster

## Máster Universitario en Ingeniería Industrial

### Diseño y desarrollo de un simulador de ventilación mecánica

#### MEMORIA

**Autor:**

Alejandro Otero Cornide

**Tutora:**

Leidy Yanet Serna Higueta

**Convocatoria:**

Septiembre 2021



Escola Tècnica Superior d'Enginyeria Industrial de Barcelona





## Agradecimientos

*Agradezco muy especialmente a la Dra. Leidy Yanet Serna Higueta, tutora de este trabajo, por su apoyo, guía y dedicación.*

*Agradecer también a la Universidad de Antioquia, por su generosidad y cortesía en su contribución a este proyecto.*

*Por supuesto, a mis padres, por haberme dado todas las oportunidades al alcance de sus manos, así como todo su apoyo.*

*Y no menos importante, a todas aquellas personas que han hecho de mi etapa universitaria la mejor hasta el momento.*

*Alejandro*



# Resumen

Los ventiladores mecánicos son equipos que proporcionan asistencia total o parcial a la ventilación del paciente, cuando este es incapaz de mantener una efectiva función respiratoria por sí mismo. Son, por tanto, dispositivos que deben cumplir con rigurosidad y robustez las especificaciones de diseño. Además, deben facilitar la configuración y monitorización por parte del personal sanitario, de cara a proporcionar el nivel de asistencia adecuado. Los simuladores de ventilación mecánica son una pieza clave tanto para el desarrollo de nuevas funcionalidades como para el entrenamiento de los profesionales clínicos. Sin embargo, el detalle del funcionamiento interno del ventilador habitualmente es escaso o nulo.

Este proyecto desarrolla un simulador de ventilación artificial basado en un modelo matemático muy detallado de un ventilador mecánico y del sistema respiratorio de un paciente. La funcionalidad de esta solución es doble. Por un lado, proporciona información precisa de cómo el flujo de aire interacciona con los distintos elementos que conforman un ventilador. Esto es de gran utilidad para ingenieros y diseñadores, los cuales pueden realizar sus propias modificaciones sobre cualquiera de los componentes del ventilador e implementar así sus propios diseños. Por el otro lado, se desarrolla una interfaz gráfica que persigue simular una experiencia real de ventilación mecánica. Esto es útil para el personal sanitario en cuanto que permite el aprendizaje y entrenamiento en los cuatro modos de ventilación implementados, pero también para los perfiles más técnicos, puesto que les facilita, además del mismo entrenamiento, un entorno donde probar sus propios diseños.

**Palabras clave:** ventilador mecánico, modelo, simulador, modos, control.



# Abstract

Mechanical ventilators are devices that provide full or partial assistance to the patient's ventilation when he or she is unable to maintain effective respiratory function on their own. They are, therefore, devices that must meet design specifications rigorously and robustly. In addition, they must facilitate configuration and monitoring by healthcare personnel to provide the appropriate level of care. Mechanical ventilation simulators are a key part of both the development of new functionalities and the training of clinical professionals. However, there is usually little or none detail on the inner workings of the ventilator.

This project develops an artificial ventilation simulator based on a very detailed mathematical model of a mechanical ventilator and a patient's respiratory system. The functionality of this solution is twofold. On one side, it provides accurate information on how the airflow interacts with the different elements that constitute a ventilator. This is specially useful to engineers and designers, who can make their own modifications to any of the ventilator's components and thereby implement their own designs. On the other side, a graphical interface is developed that aims to simulate a real mechanical ventilation experience. This is useful for healthcare personnel in that it allows learning and training in the four modes of ventilation implemented, but also for more technical profiles, as it provides them, in addition to the same training, with an environment in which to test their own designs.

**Keywords:** mechanical ventilator, model, simulator, modes, control.





# Índice general

<b>Resumen</b>	<b>VII</b>
<b>Abstract</b>	<b>VII</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Justificación . . . . .	1
1.2 Objetivos . . . . .	1
1.3 Estructura de la memoria . . . . .	2
<b>2 Estado del arte</b>	<b>3</b>
2.1 Ventiladores mecánicos . . . . .	3
2.1.1 Contexto histórico . . . . .	3
2.1.2 Fundamentos de la ventilación . . . . .	4
2.1.3 Componentes de los ventiladores . . . . .	7
2.1.4 Clasificación y modos de ventilación . . . . .	11
2.1.4.1 Sistema de control . . . . .	12
2.1.4.2 Modos de ventilación . . . . .	16
2.1.4.3 Modos de interés . . . . .	19
2.2 Entorno MATLAB . . . . .	26
2.2.1 Simulink/Simscape . . . . .	26
2.2.1.1 Modelo de ventilador mecánico y paciente . . . . .	27
2.2.2 App Designer . . . . .	30
<b>3 Desarrollo</b>	<b>33</b>
3.1 Planteamiento . . . . .	33
3.1.1 Objetivo . . . . .	33
3.1.2 Solución adoptada . . . . .	33
3.2 Modelo en Simulink . . . . .	34
3.2.1 Funcionamiento general . . . . .	34
3.2.1.1 Sistema Respiratorio . . . . .	37
3.2.1.2 Ventilador . . . . .	39
3.2.2 VC-CMV . . . . .	43
3.2.3 PC-CMV . . . . .	45
3.2.4 PSV . . . . .	47
3.2.5 PAV . . . . .	50
3.3 Interfaz gráfica . . . . .	52
3.3.1 <i>Front-end</i> . . . . .	52
3.3.2 <i>Back-end</i> . . . . .	53
3.3.2.1 Modificación del modelo . . . . .	53
3.3.2.2 Simulación y monitorización . . . . .	59
<b>4 Validación</b>	<b>65</b>
4.1 Validación cualitativa . . . . .	65

4.1.1	Modos CMV . . . . .	65
4.1.2	PAV . . . . .	66
4.2	Validación cuantitativa . . . . .	69
4.2.1	Datos experimentales . . . . .	69
4.2.2	Metodología . . . . .	70
4.2.3	Resultados y discusiones . . . . .	71
<b>5</b>	<b>Valoración económica</b>	<b>75</b>
<b>6</b>	<b>Impacto ambiental</b>	<b>77</b>
<b>7</b>	<b>Conclusiones</b>	<b>79</b>
7.1	Cumplimiento de objetivos . . . . .	79
7.2	Futuro del proyecto . . . . .	79
	<b>Glosario</b>	<b>81</b>
	<b>Bibliografía</b>	<b>83</b>



# Índice de figuras

2.1	Ventilador de presión negativa de 1838 o <i>tank ventilator</i> . . . . .	3
2.2	Pacientes de polio en <i>iron lungs</i> . . . . .	4
2.3	Ventilador de última generación de la marca Maquet . . . . .	5
2.4	Esquema de la dinámica del sistema respiratorio . . . . .	6
2.5	Dinámica de los procesos respiratorios en función de la constante de tiempo . . . . .	7
2.6	Ventilador mecánico portátil con sistema de control neumático O-Two CAREvent . . . . .	9
2.7	Curvas de salida ideales de un ventilador . . . . .	9
2.8	Esquema de los componentes básicos de un ventilador . . . . .	10
2.9	Distintas configuraciones del circuito del paciente . . . . .	11
2.10	Esquema neumático del ventilador Dräger Inifinity V500 . . . . .	12
2.11	Criterios para la determinación de las variables de fase durante una respiración asistida . . . . .	13
2.12	Comparativa entre variables objetivo y de ciclado . . . . .	14
2.13	Visualización de la PEEP intrínseca y extrínseca . . . . .	15
2.14	Esquema general de un circuito de control en lazo cerrado . . . . .	17
2.15	Esquema óptimo de control del objetivo, del modo de ventilación ASV . . . . .	18
2.16	Algoritmo para determinar la variable de control de un modo ventilatorio . . . . .	19
2.17	Algoritmo para determinar la secuencia respiratoria de un modo ventilatorio . . . . .	20
2.18	Forma de onda de la presión generada por el ventilador en el modo PC-CMV . . . . .	22
2.19	Curva de flujo en el modo PSV con una ETS del 25 % . . . . .	22
2.20	Efecto de distintos modos de ventilación en la relación entre esfuerzo instantáneo del paciente y presión entregada/consecuencias ventilatorias . . . . .	23
2.21	Diagrama y funcionamiento del modo PAV . . . . .	24
2.22	Representación esquemática de una unidad PAV móvil . . . . .	26
2.23	Modelo de partida en Simulink/Simscape de un ventilador mecánico de presión positiva y un sistema respiratorio . . . . .	27
2.24	Detalle del subsistema <i>Control Signal</i> en el modelo de partida . . . . .	28
2.25	Detalle del subsistema <i>Humidifier</i> en el modelo de partida . . . . .	28
2.26	Detalle del subsistema <i>Check Valve</i> en el modelo de partida . . . . .	29
2.27	Detalle del subsistema <i>Sensors</i> en el modelo de partida . . . . .	29
2.28	Detalle del subsistema <i>Volume Data</i> en el modelo de partida . . . . .	30
2.29	Vista Design View de App Designer con una aplicación de ejemplo . . . . .	31
3.1	Nivel superior del modelo <i>Simulator</i> . . . . .	35
3.2	Detalle del bloque <i>Variant Model</i> en el modelo <i>Simulator</i> . . . . .	36
3.3	Nivel superior del modelo que implementa el modo VC-CMV . . . . .	36
3.4	Detalle del subsistema <i>Respiratory System</i> . . . . .	37
3.5	Detalle del subsistema <i>Pmus</i> en el modelo <i>Simulator</i> . . . . .	37
3.6	Distintos tipos de señales de esfuerzo inspiratorio generadas por el subsistema <i>Pmus</i> . . . . .	39
3.7	Detalle del subsistema <i>Ventilator</i> en el modelo que implementa el modo VC-CMV . . . . .	40
3.8	Esquema de funcionamiento de una válvula PEEP en la rama espiratoria . . . . .	40
3.9	Configuración del bloque <i>Pressure to Area</i> en una <i>Check Valve</i> . . . . .	41
3.10	Relación entre la presión diferencial y el área de apertura de las <i>Check Valve</i> y <i>PEEP Valve</i> . . . . .	42
3.11	Detalle del subsistema <i>Sensors</i> en el modelo que implementa el modo PC-CMV . . . . .	43

3.12 Detalle del subsistema *Control* en el modelo que implementa el modo VC-CMV . . . . . 43

3.13 Flujos medidos en la vía aérea con un control en lazo abierto y otro en lazo cerrado en el modo VC-CMV . . . . . 44

3.14 Detalle del elemento Triggered Subsystem del subsistema Control en el modelo que implementa el modo VC-CMV . . . . . 45

3.15 Resultados de simulación del modo VC-CMV . . . . . 46

3.16 Detalle del subsistema *Ventilator* en el modelo que implementa el modo PC-CMV . . . . . 47

3.17 Detalle del subsistema *Control* en el modelo que implementa el modo PC-CMV . . . . . 47

3.18 Comparación de resultados de simulación del modo PC-CMV con control en lazo abierto y en lazo cerrado . . . . . 48

3.19 Detalle del subsistema *Control* en el modelo que implementa el modo PSV . . . . . 49

3.20 Detalle de subsistema *Expiration Trigger* en el modelo que implementa el modo PSV . . . . . 49

3.21 Resultados de simulación del modo PSV . . . . . 50

3.22 Detalle del subsistema *Control* en el modelo que implementa el modo PAV . . . . . 51

3.23 Resultados de simulación del modo PAV . . . . . 51

3.24 Panel *Patient* en la aplicación VentApp . . . . . 52

3.25 Panel *Ventilator* en modo VC-CMV en la aplicación VentApp . . . . . 53

3.26 Panel *Ventilator* en modo PC-CMV en la aplicación VentApp . . . . . 54

3.27 Panel *Ventilator* en modo PSV en la aplicación VentApp . . . . . 55

3.28 Panel *Ventilator* en modo PAV en la aplicación VentApp . . . . . 56

3.29 Panel *Monitor* en la aplicación VentApp con los resultados de una simulación en modo PSV . . . . . 57

3.30 Instalación de *VentApp* en el entorno MATLAB . . . . . 63

4.1 Curvas de flujo y presión en la vía aérea en un paciente real recibiendo ventilación en modo VC-CMV (A) y PC-CMV (B) . . . . . 66

4.2 Curvas de flujo, volumen y presión en la vía aérea obtenidas del simulador desarrollado en el modo VC-CMV . . . . . 67

4.3 Curvas de flujo, volumen y presión en la vía aérea obtenidas del simulador desarrollado en el modo VC-CMV . . . . . 67

4.4 Curvas de esfuerzo inspiratorio, presión en la vía aérea, volumen y flujo en un paciente real recibiendo ventilación en modo PSV y PAV . . . . . 68

4.5 Curvas de esfuerzo inspiratorio, presión en la vía aérea, volumen y flujo en un paciente real recibiendo ventilación en modo PAV . . . . . 68

4.6 Resultados del proceso de validación del modo PSV . . . . . 72

4.7 Curvas reales y simuladas de presión en la vía aérea, flujo y volumen de un sujeto aleatorio 73

# Índice de tablas

2.1	Generaciones de ventiladores mecánicos . . . . .	4
4.1	Parámetros de entrada al modelo en el proceso de validación . . . . .	70
5.1	Valoración económica del proyecto . . . . .	75



## Introducción

### 1.1. JUSTIFICACIÓN

El presente proyecto surge de la necesidad de acercar los fundamentos de la ventilación mecánica tanto al personal clínico como también a profesionales de la ingeniería y de otras disciplinas técnicas relacionadas. El proyecto persigue, por tanto, aportar conocimientos y una infraestructura de aprendizaje no solo sobre las bases fisiológicas de la ventilación artificial, sino también sobre el funcionamiento de los propios ventiladores mecánicos.

Algunos simuladores de ventilación artificial modelan el ventilador mecánico como una caja negra, la cual produce, ante determinadas entradas, las salidas que se esperarían de un ventilador real completamente funcional. Durante este proceso, sin embargo, los detalles de cómo el flujo de aire interacciona con el circuito ventilatorio son escasos o nulos. Si bien estos detalles no son de especial utilidad para el personal sanitario, sí tienen relevancia para los profesionales técnicos diseñadores de estos dispositivos. De cara a obtener un conocimiento suficiente sobre el diseño de ventiladores mecánicos, es necesario que los ingenieros también comprendan las consecuencias clínicas de la ventilación artificial, las maneras de proporcionarla y cómo el paciente interactúa con esta, por lo que aunar ambas ciencias se vuelve una necesidad.

En torno a esta idea se fundamenta la solución adoptada en este trabajo. Partiendo de un modelo matemático que reproduce con gran nivel de detalle los diferentes componentes físicos tanto de un ventilador mecánico como del sistema respiratorio de un paciente, se desarrolla un simulador que permite recrear una experiencia real de ventilación artificial. Este simulador implementa cuatro modos ventilatorios distintos, desde los más elementales a alguno de los más novedosos, pasando por uno de los más extendidos, lo cual facilita una introducción progresiva y de utilidad al conocimiento de las diferentes estrategias de ventilación. Para facilitar este aprendizaje, se desarrolla una interfaz gráfica que busca imitar a un ventilador mecánico real. El modelo matemático se ejecuta de forma opaca para el usuario, en segundo plano; sin embargo, este siempre se encuentra accesible, y cualquier modificación que se le realice se vería reflejada a través de la interfaz gráfica.

### 1.2. OBJETIVOS

En base a la necesidad mencionada, se establece el objetivo principal de este proyecto, que es el de diseñar un modelo virtual de un ventilador mecánico que reproduzca fielmente el comportamiento real de uno de estos dispositivos. Además, el modelo debe cumplir dos funciones:

- Servir como punto de partida para ingenieros y técnicos, de forma que puedan simular sus propios diseños de un ventilador mecánico partiendo de un modelo ya probado y con una infraestructura funcional.
- Permitir a los usuarios interactuar con él de forma sencilla, de modo que simule una experiencia de ventilación mecánica real y sirva como medio de aprendizaje tanto para personal clínico como para ingenieros y técnicos.

Para alcanzar estos propósitos, se fijan una serie de objetivos parciales a lo largo del proyecto hasta la consecución del resultado tangible:

1. Estudio de los fundamentos de la ventilación artificial, modos ventilatorios y funcionamiento de los ventiladores mecánicos.
2. Estudio del modelo de partida, sus componentes y funcionamiento, así como del entorno Simulink.
3. Planteamiento de la solución del modelo, modos a implementar y método, considerando la integración con la interfaz gráfica.
4. Desarrollo del modelo en Simulink para los cuatro modos de ventilación.
5. Validación de los modelos.
6. Planteamiento de la solución de la interfaz gráfica.
7. Desarrollo de la interfaz gráfica.
8. Validación de la solución global.

### 1.3. ESTRUCTURA DE LA MEMORIA

El presente documento es un reflejo de los conocimientos tanto adquiridos como desarrollados a lo largo de la consecución de estos objetivos. Además del presente capítulo introductorio, se incluyen los enumerados a continuación:

2. *Estado del arte*, donde se recogen los conceptos teóricos más relevantes para el alcance del proyecto, de forma que se ponga en contexto al lector y se facilite así el entendimiento de la solución desarrollada. Forma parte, a su vez, del valor añadido del proyecto, dado el carácter didáctico del mismo.
3. *Desarrollo*, en el cual se recoge y detalla la aportación tangible obtenida de la realización de este proyecto.
4. *Validación*, que describe el proceso y los resultados que legitiman el correcto funcionamiento de la solución desarrollada.
5. *Valoración económica*, donde se recogen los costes desprendidos de la realización de este trabajo.
6. *Impacto ambiental*, que incluye la evaluación de los efectos generados por la solución desarrollada en materia de impacto ambiental.
7. *Conclusiones*, donde se recogen las ideas principales en relación a los resultados obtenidos, tanto puntos fuertes como carencias a suplir con trabajos futuros.
8. *Glosario*, en el que se recogen los términos más utilizados a lo largo del proyecto, con su correspondiente definición para facilitar al lector el entendimiento del proyecto.
9. *Bibliografía*, donde se listan las diferentes fuentes documentales utilizadas en la elaboración de este trabajo.



# Estado del arte

Los avances en el estudio de la respiración han estado siempre ligados a los descubrimientos tanto de anatomistas, químicos y fisiólogos como incluso de exploradores, que a lo largo de la historia han sometido a su sistema respiratorio a condiciones extremas. Ya los antiguos griegos tenían su propia concepción sobre el objetivo de esta función, inherente a todos los seres vivos. En los *Tratados Hipocráticos*, conjunto de escritos médicos atribuidos a Hipócrates, padre de la medicina contemporánea, declara que el propósito de la respiración es el de refrigerar el corazón [1].

En cuanto a los orígenes de la ventilación artificial, podríamos ubicarlos allá por el siglo XVI cuando, mediante la inflación de aire directamente en la tráquea, el médico belga Vesalius consiguió paliar el neumotórax en las vivisecciones de animales y conseguir así mantener al sujeto con vida [2]. Además de esta hazaña, se dice que consiguió, también mediante ventilación artificial, hacer latir de nuevo el corazón de un fallecido aristócrata español, lo cual casi le cuesta la vida a manos de la Santa Inquisición.

Si bien sus descubrimientos fueron olvidados durante muchos años, hoy en día la ventilación mecánica es una técnica que está a la orden del día en todos los hospitales del mundo, y los avances y mejoras alrededor de esta práctica están lejos de cesar.

## 2.1. VENTILADORES MECÁNICOS

### 2.1.1. Contexto histórico

El uso de la ventilación mecánica comenzó a extenderse a lo largo del siglo XIX y primera mitad del siglo XX con el uso de ventiladores de presión negativa [3]. La primera descripción que se tiene de un ventilador mecánico (VM) de este tipo es del 1838, por John Dalziel, un ventilador de cuerpo entero en forma de tanque en el que el paciente se sentaba con tan solo la cabeza fuera de este (Fig. 2.1).

Pero no sería hasta mediados del siglo XX cuando la poliomielitis (más conocida como la polio) popularizó el uso del *iron lung*, llegando las UCIs de la época a tratar a docenas de pacientes al mismo tiempo con esta nueva forma de ventilación mecánica por presión negativa (Fig. 2.2)

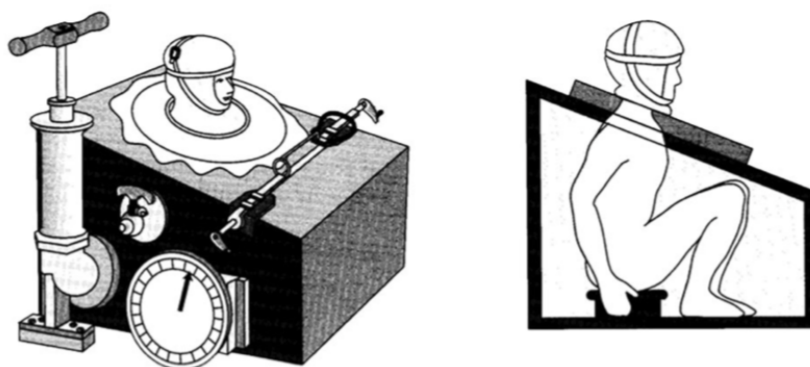


Fig. 2.1: Ventilador de presión negativa de 1838 o *tank ventilator* [3].



Fig. 2.2: Pacientes de polio en *iron lungs* en el Auditorio de Rancho Los Amigos, California, 1953 [4].

Actualmente, los VMs que se encuentran en los hospitales distan bastante de parecerse a estos primitivos modelos. Por una parte, porque los actuales utilizan presión positiva para ventilar, lo cual les otorga una fisonomía completamente distinta a los del tipo «pulmón de hierro»; y por otra, si bien la ventilación por presión positiva no es algo nuevo (véase Vesalius), por su integración de la electrónica digital, que ha conseguido propiciar la cuarta generación de ventiladores mecánicos (véase Tabla 2.1) (Fig.2.3).

Generación	Años	Característica distintiva
Primera	Principios del siglo XX hasta mediados de los 1970s	Solo ventilación controlada por volumen
Segunda	Mediados de los 1970s hasta principios de los 1980s	Primera aparición de inspiración disparada por paciente
Tercera	Principios de los 1980s hasta finales de los 1990s	Control con microprocesador
Cuarta	Finales de los 1990s hasta el presente	Multitud de modos de ventilación
Future	Solo el tiempo lo dirá	Ventilación inteligente con apoyo a la decisión

Tabla 2.1: Generaciones de ventiladores mecánicos [3]

### 2.1.2. Fundamentos de la ventilación

Antes de entrar en detalle sobre los ventiladores mecánicos, es preciso profundizar un poco sobre la ventilación mecánica en sí misma. Esta técnica puede definirse como «un método físico que utiliza un aparato mecánico para el soporte artificial de la ventilación y la oxigenación, cuando el sistema respiratorio es insuficiente» [6]. Se dice que un paciente presenta insuficiencia respiratoria (IR) cuando



Fig. 2.3: Ventilador de última generación de la marca Maquet® [5]

el nivel de oxígeno en sangre es demasiado bajo o el de dióxido de carbono demasiado alto, o ambos [7]. Esto puede estar causado por diferentes afecciones, siendo una de ellas la COVID-19; de ahí la importancia de esta técnica en el tratamiento de esta pandemia que, aún a día de hoy, sigue azotando nuestra sociedad.

Sin embargo, en el proceso de la respiración, el VM no sustituye por completo al sistema respiratorio humano, puesto que no es capaz de asegurar el intercambio molecular de oxígeno y dióxido de carbono alveolo-capilar (respiración), sino solamente el movimiento de gases dentro y fuera de los pulmones (ventilación). Es por esto que el uso extendido del término «respirador» como sinónimo de ventilador no es del todo correcto [6].

Queda claro, por tanto, que la función del VM es la de mantener un flujo adecuado de gases entre el interior de los pulmones y el entorno. Para ello, como se ha introducido anteriormente, los ventiladores pueden ser de presión negativa o positiva.

En ambos casos, el objetivo es generar un gradiente de presión positivo entre la vía aérea superior y la superficie corporal para producir la inspiración. Mientras que en el primer caso (Fig.2.1 y 2.2) esto se consigue disminuyendo la presión alrededor de la caja torácica (de ahí que el cuerpo deba estar dentro del tanque), en el segundo caso se aumenta la presión en la vía aérea, bien sea de forma invasiva — mediante la colocación de un dispositivo en la tráquea— o no invasiva —mediante una interface fuera de la vía aérea— [8].

Este gradiente, que se denomina presión transrespiratoria, es la resultante de todas las presiones que actúan sobre el sistema respiratorio —véase la presión generada por el ventilador y la presión muscular, generada por la contracción de los músculos respiratorios—. Además, tiene que ser suficiente como para vencer la impedancia del sistema respiratorio, formada por una componente resistiva y otra elástica. La componente resistiva, o presión transviaaérea, es el gradiente de presión entre la vía aérea superior y los alvéolos, y viene determinada por la resistencia de la vía aérea y por el flujo inspiratorio. La componente elástica, o presión transtorácica, es el gradiente de presión entre el espacio alveolar y la superficie corporal, determinada por el volumen circulante y la distensibilidad toracopulmonar (véase Fig. 2.4 ) [6].

Esta dinámica se puede expresar mediante un balance de fuerzas de acuerdo a la tercera ley de Newton (principio de acción y reacción), gracias a la «ecuación del movimiento del sistema respiratorio» (2.1) [9].

$$P_{TR} = P_R + P_E \tag{2.1}$$

Que a su vez se puede expresar de la siguiente forma:

$$P_{vent}(t) + P_{mus}(t) = R\dot{V}(t) + EV(t) \tag{2.2}$$

Donde  $P_{vent}$  es la presión generada por el ventilador,  $P_{mus}$  la generada por los músculos respiratorios, y  $V$  y  $\dot{V}$  son, respectivamente, el volumen y el flujo. Los términos  $R$  y  $E$  hacen referencia a la impedancia del sistema respiratorio (resistencia y elastancia, respectivamente). Nótese que, en estas ecuaciones, se utiliza el parámetro de elastancia en lugar de la distensibilidad o compliancia ( $C$ ), que es la magnitud inversa; mientras que la primera es una medida de la capacidad de los pulmones de recuperar su forma inicial, la compliancia relaciona el cambio de volumen ante un aumento de la presión.

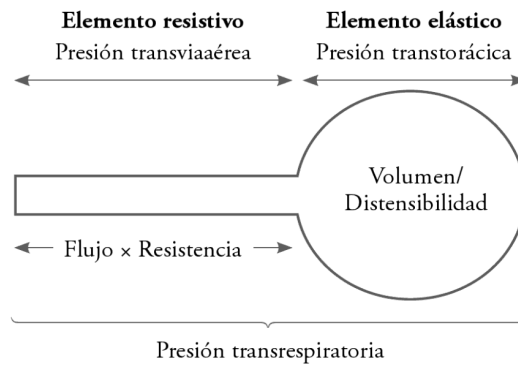


Fig. 2.4: Esquema de la dinámica del sistema respiratorio [6]

Mencionar que, si bien esta representación matemática del sistema respiratorio es la más extendida y utilizada, otros modelos más complejos han sido objeto de estudio, como el modelo de doble compartimento [10].

De la ecuación 2.2 se pueden deducir varias cuestiones. Por una parte, que si la  $P_{mus}$  es suficiente como para generar el flujo inspiratorio, la  $P_{vent}$  no es necesaria (caso de la respiración natural espontánea); y, por la otra, que en casos de apnea o de parálisis muscular, la  $P_{mus}$  es nula y todo el trabajo de la ventilación recaería sobre el VM. A mayores, también se puede observar que ambos términos pueden complementarse o que, en caso de asincronía paciente-ventilador, la  $P_{TR}$  resultante disminuiría y el flujo y volumen entregados sería menor. Si, además, hacemos nulos tanto  $P_{mus}$  como  $P_{vent}$ , obtenemos la expresión de una espiración pasiva:

$$- R\dot{V}(t) = EV(t) \tag{2.3}$$

El signo negativo en (2.3) indica el sentido del flujo correspondiente a la espiración, y cuya magnitud es función de la energía almacenada en el compartimento elástico durante la inspiración [9].

Por último, se puede observar que la ecuación del movimiento (2.2) es una función diferencial de primer orden, que se puede expresar en el dominio de Laplace tal que [11]:

$$\frac{V(s)}{P_{TR}(s)} = \frac{1/E}{\frac{R}{E}s + 1} \tag{2.4}$$

Donde  $1/E$  es la ganancia estática y  $R/E$  la constante de tiempo del sistema ( $\tau$ ), medida en segundos. Así, frente a una entrada de tipo escalón, como puede ser una espiración pasiva o una inspiración a

presión constante, la respuesta del sistema respiratorio es del tipo exponencial (véase Fig. 2.5). Este concepto es muy importante sobre todo en el proceso de la espiración pasiva, cuya duración debe ser siempre de al menos  $3\tau$  para evitar atrapamientos de gas severos y con ello una elevada auto-PEEP. La auto-PEEP o PEEP intrínseca es la diferencia al final de la espiración entre la presión positiva alveolar y la presión en la vía aérea (PEEP extrínseca, habitualmente PEEP). Si bien la auto-PEEP es un fenómeno a vigilar —aunque en determinadas ocasiones puede ser inevitable y hasta deseable, como en la ventilación de alta frecuencia—, la PEEP extrínseca es un parámetro configurable en muchos VMs por aportar ciertos beneficios clínicos [12] [13].

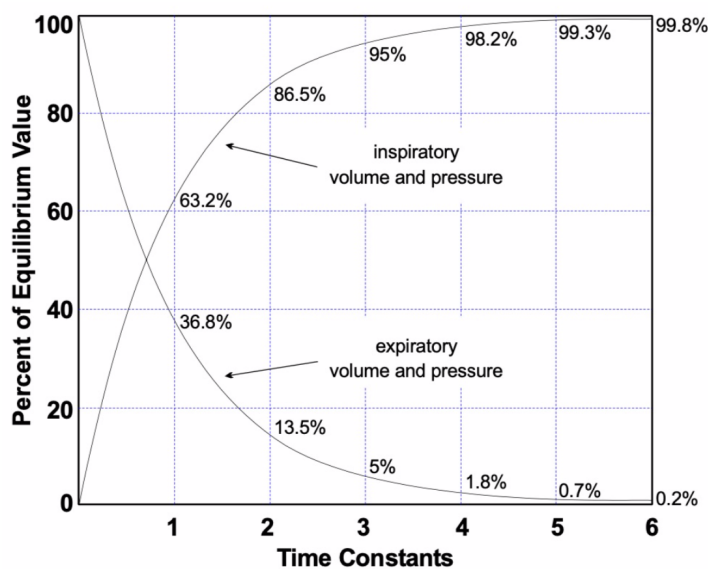


Fig. 2.5: Dinámica de los procesos respiratorios en función de la constante de tiempo [11]

### 2.1.3. Componentes de los ventiladores

Si bien existen en el mercado algunos ventiladores modernos de presión negativa, su uso es prácticamente despreciable en comparación con los de presión positiva, por lo que en adelante este trabajo se centrará en el estudio de estos últimos, por ser estos los más extendidos y debido a la idiosincrasia del presente proyecto.

Con esto en mente, se pueden clasificar los elementos básicos de un ventilador mecánico comercial de acuerdo a la siguiente estructura [12] [14]:

- **Entradas.** Los ventiladores normalmente se alimentan mediante dos tipos de energía:
  - **Eléctrica**, que a su vez puede entregarse en corriente alterna (a 230 V y 50 Hz en la Unión Europea) directamente de las tomas de los edificios o en corriente continua (típicamente a 10-30 V) a partir de baterías, incluidas principalmente en VMs transportables y como medida de seguridad en casos de desconexión de la red eléctrica. Este tipo de energía se utiliza tanto para alimentar los compresores de aire —que hacen funcionar los circuitos neumáticos y entregan el aire al paciente— como para, mediante la conversión a corriente continua si es el caso, la alimentación de los componentes electrónicos del ventilador.
  - **Neumática**, obtenida a través de los suministros de gas comprimido en las paredes de los hospitales o de depósitos. Utilizada para inflar los pulmones en gran parte de los ventiladores, que incluyen válvulas reductoras de presión para adaptar las características del gas a las condiciones de operación del VM.

Alternativamente, algunos ventiladores también utilizan esta fuente de energía para sus mecanismos de control, prescindiendo por completo de la energía eléctrica. Esta característica los hace ideales para situaciones en las que la electricidad no siempre está disponible, como método de respaldo en caso de fallo de esta o particularmente en entornos en los que la corriente eléctrica no es deseable, como en las proximidades de equipos de resonancia magnética.

- **Conversión y control.** Estos dos conceptos están estrechamente ligados y son los responsables de convertir la energía de entrada en una salida predeterminada de flujo y presión. Aún así, podemos distinguir:

- **Mecanismo de accionamiento.** Es el elemento encargado de convertir la energía de entrada en trabajo útil, y determina en parte la salida que produce el ventilador. Este mecanismo puede consistir bien en una aplicación directa del gas comprimido tras pasar por una válvula reductora de presión, en el caso de los VMs alimentados neumáticamente, o bien, en el caso de alimentación únicamente eléctrica, en una aplicación indirecta utilizando alguno de los elementos siguientes:
  - Un *compresor*, diseñado para mover un flujo de gas relativamente bajo a espacios de más alta presión. Se suelen utilizar en ventiladores de cuidados intensivos, puesto que son de tamaño y consumo eléctrico considerable.
  - Un *soplador*, diseñado para mover flujos de gas mayores con un incremento de presión limitado. En contraposición con los compresores, estos se encuentran más a menudo en VMs pequeños y portátiles.
- **Válvula de control de flujo.** Estos elementos permiten controlar el flujo generado por el mecanismo de accionamiento y que se entregará al paciente. En su versión más sencilla, esta consiste en una válvula simple con una apertura fija que permite entregar un flujo constante, y que encuentra su uso principal en pequeños ventiladores transportables. Otras válvulas ajustables manualmente se han utilizado en el pasado, pero hoy en día, con el avance y el abaratamiento de la microelectrónica, la mayoría de VMs incluyen sistemas más o menos complejos que, mediante dispositivos lógicos, permiten controlar estas válvulas para modular de múltiples maneras las curvas de presión, flujo y volumen entregados.

Además, el control de esta válvula requiere de una coordinación con la válvula espiratoria. En el caso más sencillo, cuando la primera se abre para iniciar la inspiración, la segunda se cierra para que el total del flujo de gas sea dirigido hacia el paciente; y a la inversa, cuando comienza la espiración, la válvula de control de la salida se cierra para interrumpir el flujo hacia el paciente y la válvula espiratoria se abre para permitir el vaciado de los pulmones. En los VMs más sofisticados, complejas formas de interacción entre ambas válvulas permiten una gran variedad de formas de onda de presión, volumen y flujo que maximizan la sincronización entre el ventilador y el paciente.

- **Sistema de control.** Precisamente en esta clasificación se incluyen todos los elementos que permiten el control de las válvulas de flujo y de exhalación, de acuerdo al modo de ventilación y características de la salida deseadas.

Aunque previamente se ha mencionado el uso extendido de los sistemas de control basados en la electrónica digital, estos también pueden tener su fundamento en componentes mecánicos, neumáticos (Fig. 2.6) o que utilicen otro tipo de fluido. En el caso de los mecánicos —en los primeros VMs operados manualmente—, estos pueden ser palancas, poleas, levas, etc., mientras que en sistemas de control que utilizan un fluido, este operará elementos como membranas o pistones; pero además, el avance en el uso de los láseres ha permitido la creación de microcanales para la construcción de sistemas de control miniaturizados operados con fluido. Estos circuitos son análogos a los circuitos electrónicos, pues en ellos se usan flujos muy pequeños del fluido para generar señales y operar elementos como interruptores o temporizadores. Este tipo de sistemas de control son ideales para entornos con interferencias



Fig. 2.6: Ventilador mecánico portátil con sistema de control neumático O-Two CAREvent® [15]

electromagnéticas, pero son mucho menos comunes que los electrónicos, los cuales ofrecen mucha más versatilidad a la hora de dar forma a la salida, monitorizar las señales y programar los diferentes modos de ventilación. Estos modos son los que definen la interacción del ventilador con el paciente, y su discusión ocupará un epígrafe aparte más adelante.

- **Salidas.** Como se ha introducido anteriormente, el objetivo del control de las válvulas de flujo es el de otorgar las curvas deseadas de **presión, flujo y volumen** a la salida del ventilador. Estas formas de onda son de gran importancia ya que proporcionan una información muy valiosa sobre el estado del paciente y su progreso y el desempeño del VM.

En aquellos ventiladores que cuentan con una *interfaz gráfica*, estas tres señales suelen visualizarse en conjunto, con el eje horizontal representando la misma línea temporal y en el eje vertical la magnitud de cada una de ellas. A partir de la ecuación del movimiento del sistema respiratorio (2.2) y de la curva de una de las tres variables, se obtienen las formas de onda de las otras dos señales, dependientes de la primera. En la Fig. 2.7 se pueden observar algunos de los patrones más típicos que producen los VMs. Sin embargo, estas curvas son ideales, obtenidas de expresiones matemáticas; los ventiladores reales están sometidos a perturbaciones, ruido, y al comportamiento de la propia planta, con su correspondiente resistencia y compliancia. Aun así, estas curvas son utilizadas por los sistemas de control de los ventiladores para conseguir una aproximación de la salida deseada.

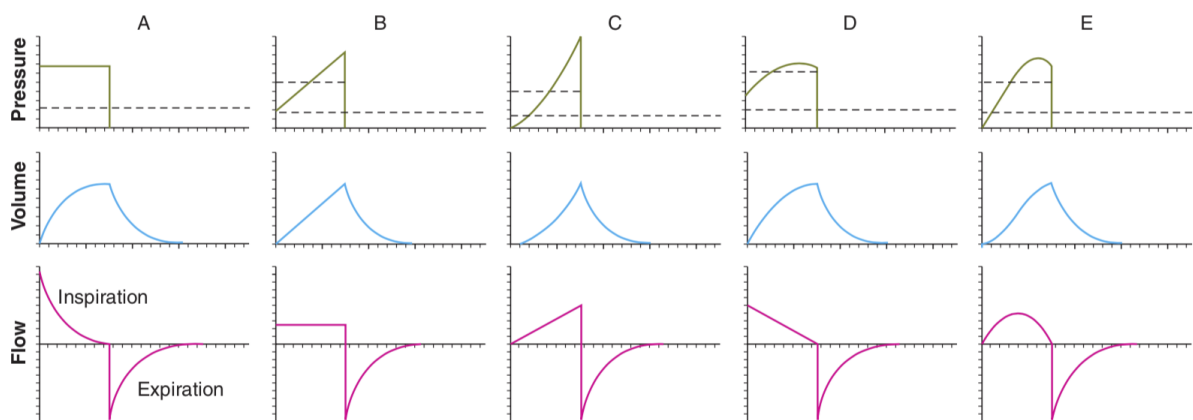


Fig. 2.7: Curvas de salida ideales de un ventilador [14]

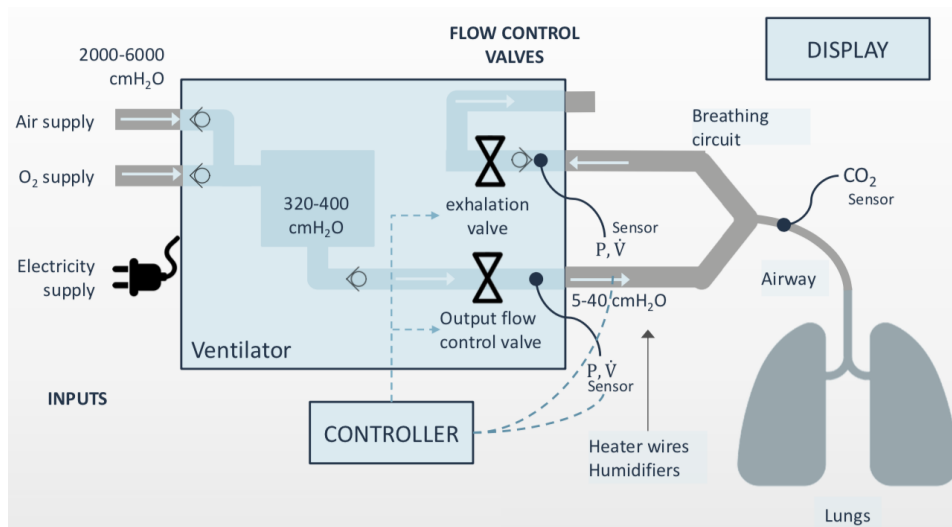


Fig. 2.8: Esquema de los componentes básicos de un ventilador [11]

Como último elemento del ventilador, se encuentra el encargado de hacer llegar el flujo generado al sistema respiratorio de la persona ventilada: el *circuito del paciente* (véase Fig. 2.8). Estos circuitos deberán cumplir las características definidas por el fabricante del VM, como pueden ser la longitud, colectores de agua, sensores (flujo, temperatura, etc.) sistemas de humidificación, filtros o conexión a un nebulizador [8].

En cuanto a la ventilación mecánica invasiva (VMI), en los que el sujeto se encuentra intubado, el circuito del paciente contará siempre con dos ramas (o corrugados) —una para el flujo inspiratorio y otra para el espiratorio— unidas por una pieza en Y. En el caso de la ventilación no-invasiva (VMNI), en la que la interfaz usada entre ventilador y paciente habitualmente es una máscara, es posible encontrarse con tan solo una rama, para el flujo inspiratorio, mientras que la espiración se realiza a través de la máscara o de una válvula espiratoria próxima a ella (véase Fig. 2.9)

Por contra, el circuito del paciente contribuye en gran parte a las diferencias que se encuentran entre las curvas ideales de presión, flujo y volumen deseadas y las entregadas realmente al paciente. Esto es debido a la propia impedancia del circuito. Así, se producirá una caída de presión entre la salida inmediata del ventilador y la apertura de la vía aérea debido a la resistencia del circuito; del mismo modo que el flujo y volumen a través de la válvula espiratoria será mayor que los entregados al paciente debido a la compliancia.

Estableciendo una analogía entre este sistema y uno eléctrico, y considerando que se pueden analizar la resistencia y la compliancia por separado, podría decirse que las resistencias del circuito y del paciente se encuentran conectadas en serie, puesto que el flujo que las atraviesa sería el mismo mientras que la caída de presión sería proporcional al valor de cada resistencia (nótese que el flujo se identifica con la intensidad y la presión con el potencial). En cuanto a la compliancia del circuito del paciente, esta es una combinación de la propia compliancia del tubo y de la compresibilidad del gas dentro de él. Ignorando la pérdida de presión debido a la resistencia del circuito (de hecho, en la práctica, esta es considerablemente menor que la de la vía aérea), la presión en el circuito es la misma que en el sistema respiratorio; no así el cambio de volumen que experimenta cada uno, debido a las distintas compliancias. Como ambos sistemas experimentan incrementos de volumen distintos en un mismo tiempo inspiratorio, se puede deducir que el flujo en ellos también es distinto ( $\text{flujo} = \text{volumen}/\text{tiempo}$ ). Nuevamente, por analogía con el sistema eléctrico, se puede deducir que las compliancias de ambos componentes están conectadas en paralelo, pues están sometidas a la misma presión pero a diferentes flujos, por lo que la compliancia total del sistema circuito-paciente es mayor que cada una de ellas.



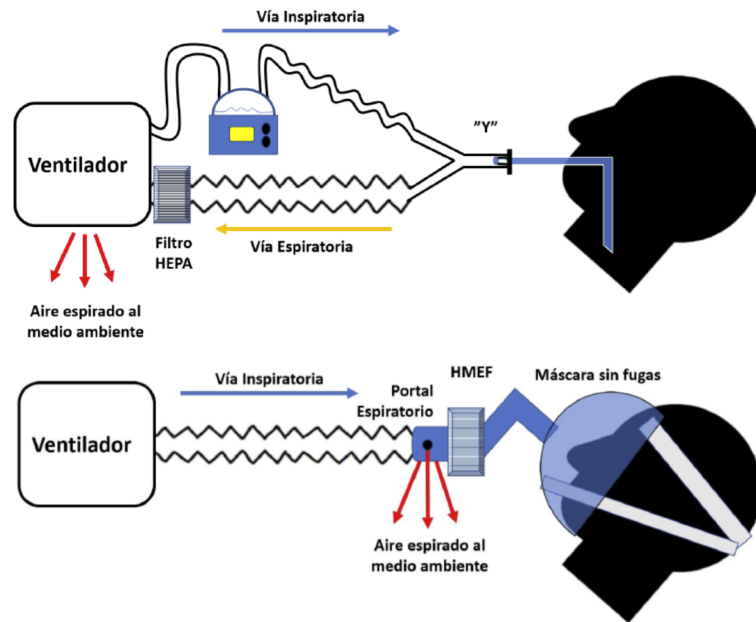


Fig. 2.9: Distintas configuraciones del circuito del paciente. Circuito de dos ramas en VMI y sistema de humidificación activa con base calefactora y filtro HEPA al final de la vía espiratoria (arriba). Circuito de una rama en VMNI con filtro HMEF entre portal espiratorio o válvula espiratoria proximal e interfaz sin fuga del paciente (abajo). [16]

La compliancia del circuito sí que puede suponer un gran impacto en la ventilación entregada, pues puede llegar a ser, en casos de neonatos, hasta 3 veces mayor que la del sistema respiratorio. En modos de ventilación en los que se pretende entregar un volumen fijo por respiración, esta puede verse afectada considerablemente si se carece de una monitorización del flujo en la vía aérea del paciente, tal como se muestra en (2.5), donde  $V_{entregado}$  es el volumen tidal ( $V_T$ ), o por respiración, recibido por el paciente,  $V_{configurado}$  es el  $V_T$  definido en los parámetros del ventilador y que se pretende alcanzar, y  $C$  y  $C_{CP}$  son, respectivamente, la compliancia pulmonar y del circuito del paciente.

$$V_{entregado} = \frac{V_{configurado}}{1 + (C_{CP}/C)} \quad (2.5)$$

En los modos en los que la consigna es una presión constante, las compliancias elevadas pueden aumentar considerablemente el tiempo de subida, lo cual disminuiría el valor máximo del flujo y el  $V_T$  ligeramente. [14]

Si bien estos son los componentes básicos de un VM, los más modernos son, al igual que ocurre con los vehículos, una auténtica caja negra donde se agolpan sistemas complejos que, pese a suponer un gran número de avances en cuanto a seguridad y confort, privan cada vez más al usuario medio de llevar a cabo personalmente ciertos mantenimientos. A modo ilustrativo, véase la Fig. 2.10, donde se muestra el complejo sistema neumático de un ventilador de cuidados intensivos.

#### 2.1.4. Clasificación y modos de ventilación

Un buen método para clasificar los distintos VMs que existen actualmente es hacerlo según su forma de operar, sin detallar demasiado pero sí lo suficiente como para poder distinguir unos de otros. Este funcionamiento característico de cada ventilador y su interacción con el paciente es lo que se denomina

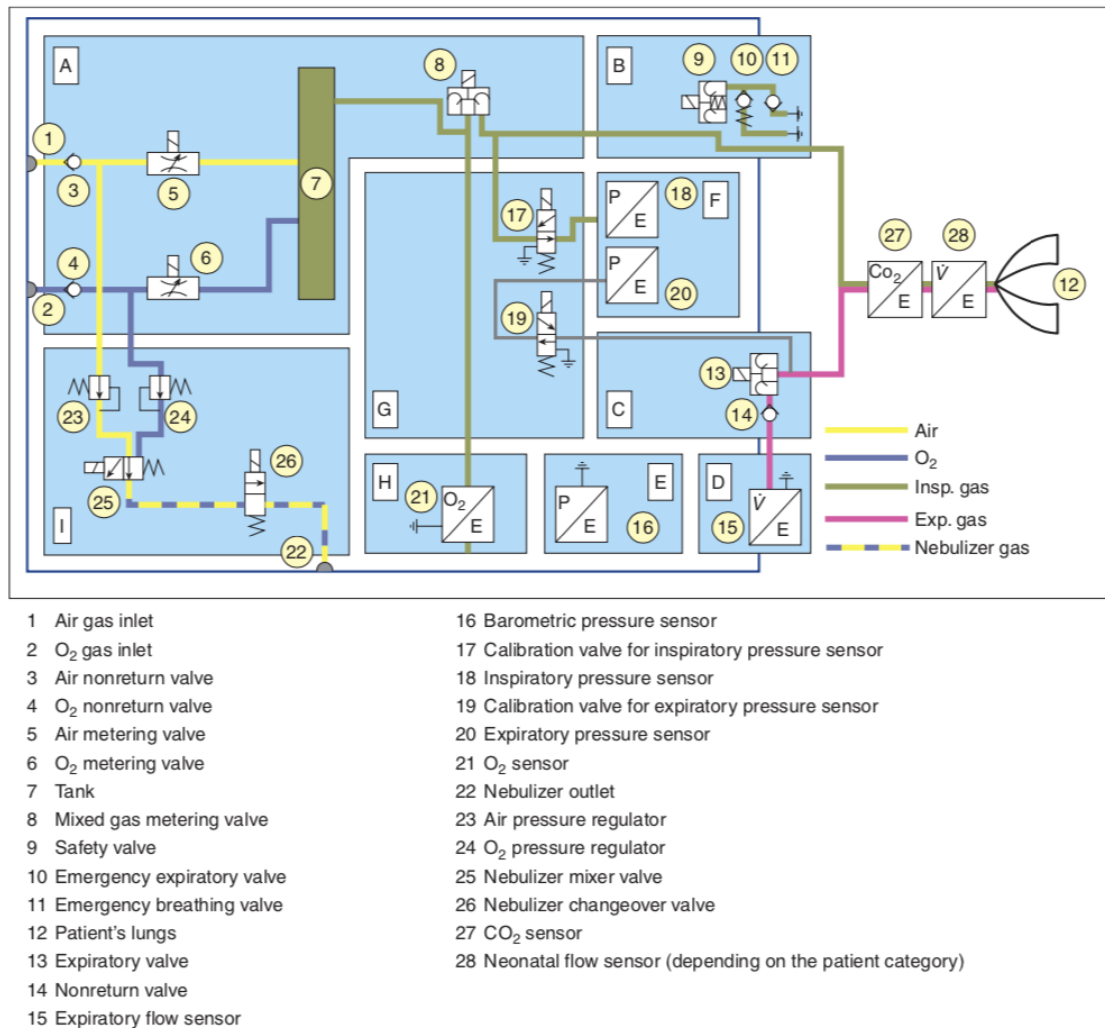


Fig. 2.10: Esquema neumático del ventilador Dräger Inifinity V500®. A. Conjunto de mezcla y medición de gas. B. Unidad inspiratoria. C. Unidad Espiratoria. D. Sensor de flujo espiratorio. E. Sensor de presión barométrico. F. Conjunto de medida de presión. G. Conjunto de calibración. H. Sensor de oxígeno. I. Conjunto de nebulizador de medicamentos. [14]

*modo de ventilación*; no obstante, un mismo VM puede disponer de varios modos en su configuración, con sus distintos parámetros, pero solo uno de ellos será el que esté en funcionamiento cada vez [9].

#### 2.1.4.1. Sistema de control

Antes de entrar al método de clasificación de los ventiladores, es necesario definir el funcionamiento del sistema de control de los VMs, y sobre todo de las **variables** que controlan:

- **Variables de control.** Habiendo introducido la ecuación del movimiento del sistema respiratorio (2.2), y tal como se ha pincelado en el epígrafe anterior, es posible predeterminar una de las tres variables (presión, flujo o volumen) convirtiéndola en la variable independiente y, por consiguiente, a las otras dos en los términos dependientes. Se puede hablar, entonces, de ventilación controlada por presión, por volumen o por flujo. En el primer caso, por ejemplo, y en caso de que el sistema respiratorio del paciente fuera pasivo ( $P_{mus} = 0$ ), las curvas de volumen y de flujo dependerían

únicamente de la forma de onda de la presión y la impedancia del sistema respiratorio (caso ideal, véase, por ejemplo, Fig. 2.7-A). Lo mismo aplica al control del volumen y del flujo, salvo por que, al ser el flujo la magnitud derivada del volumen, el control de ambas variables es redundante, por lo que a partir de ahora tan solo se considerarán variables de control **la presión y el volumen**.

Sin embargo, existen dos excepciones que podrían enmarcarse fuera de esta clasificación: la ventilación percusiva interpulmonar y la ventilación oscilatoria de alta frecuencia. En ambos casos, la variable controlada es la duración de los pulsos de flujo, por lo que tanto la presión en la vía aérea, como el flujo y el volumen inspirado dependen de los valores instantáneos de resistencia y compliancia. Podría hablarse entonces de ventilación controlada por tiempo. Destacar también que, si bien se intuye que solo es posible controlar una variable a la vez, los sistemas de control pueden alternar entre una y otra variable a lo largo de distintas respiraciones o durante una misma inspiración.

- **Variables de fase.** Dado que la respiración es un evento periódico, el ventilador tiene que ser capaz de controlar diversas variables durante el ciclo respiratorio. Una convención muy extendida es la de dividir el tiempo del ciclo respiratorio en cuatro fases: el cambio de espiración a inspiración, la inspiración, el cambio de inspiración a espiración y la espiración. Esta es muy útil para analizar cómo el VM empieza, mantiene y finaliza una inspiración, y qué hace entre ellas. En este contexto, se utiliza una variable particular para empezar, mantener y finalizar cada fase; estas variables de fase son **la presión, el flujo, el volumen y el tiempo**. En la Fig. 2.11 se muestran los criterios para determinar las distintas variables de fase durante una respiración asistida.
- **Variable de disparo.** La inspiración en una ventilación asistida **comienza** porque una de las variables de fase alcanza un valor predefinido, la variable de disparo o *trigger*. Cuando el tiempo es la variable de disparo, la inspiración comienza de acuerdo a una frecuencia respiratoria prefijada, independientemente del esfuerzo espontáneo del paciente. La presión dispara la respiración cuando el ventilador mide una caída en la presión de base debido a un esfuerzo inspiratorio del paciente,

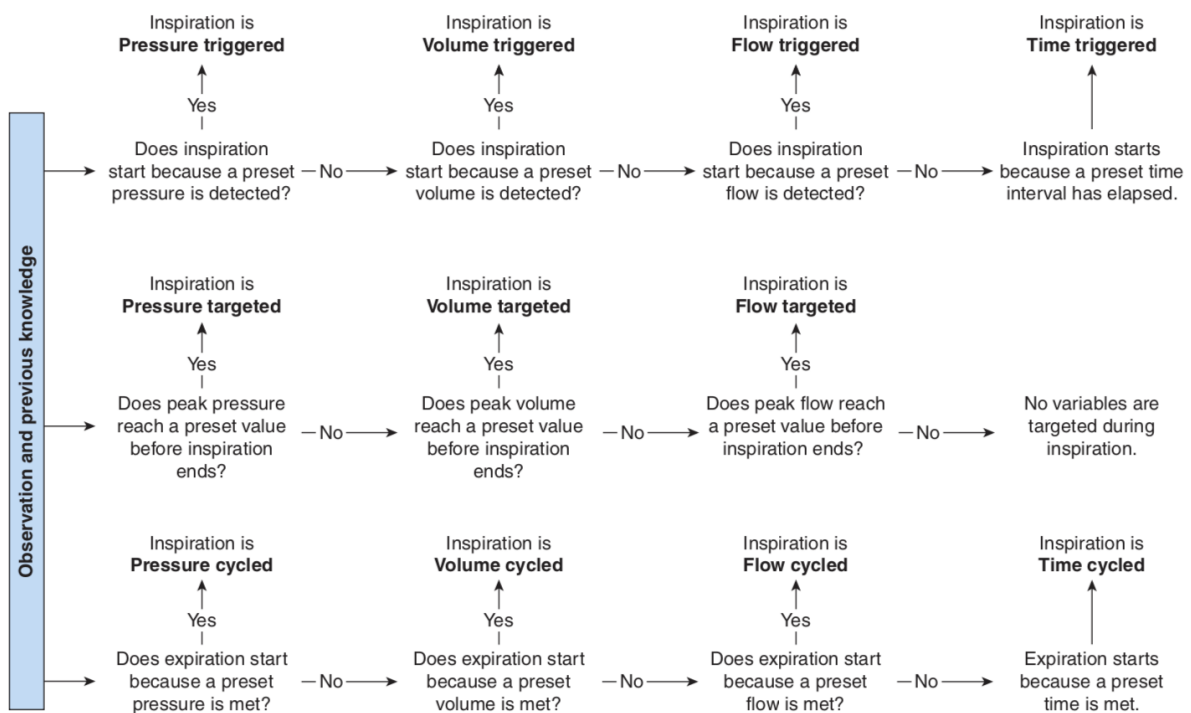


Fig. 2.11: Criterios para la determinación de las variables de fase durante una respiración asistida [9]

sin tener en cuenta la frecuencia respiratoria fijada, y del mismo modo actúa el VM cuando las variables de disparo son el flujo o el volumen.

El disparo mediante flujo es preferible frente al de presión en cuanto al trabajo que el paciente debe realizar. Esto es debido a que el trabajo es proporcional al volumen inspirado por el cambio de presión. Dado que si la variable de disparo es la presión, es necesario que haya un cambio en esta y, por tanto, un trabajo; no así si se dispara la respiración por flujo o volumen, pues la presión no necesita cambiar y, por tanto, el paciente no necesita realizar ningún trabajo, teóricamente.

- **Variable objetivo.** Con objetivo, o *target*, se entiende la **restricción** de la magnitud de una de las variables durante la inspiración. Así pues, una variable objetivo es aquella que puede alcanzar un determinado nivel y mantenerlo antes de que la inspiración termine. Es importante notar que esta variable **no finaliza** la inspiración. Variables *target* pueden ser la presión, el flujo o el volumen, o incluso todas ellas en una misma respiración. El tiempo, por contra, no puede ser una variable objetivo, puesto que establecer un límite al tiempo de inspiración causaría la terminación de esta y el comienzo de la fase espiratoria.
- **Variable de ciclado.** Del inglés *cycle variable*, esta es la que determina el **final de la inspiración** cuando su valor alcanza un nivel predefinido. Puede ser cualquiera de las variables de fase, aunque en algunos ventiladores el ciclado puede ser manual.

El ciclado por presión es más habitual en ajustes de alarmas, interrumpiendo la entrega de flujo cuando el valor de presión es demasiado alto y puede resultar peligroso. Cuando la variable de ciclado es el volumen, la espiración comienza una vez se ha entregado el  $V_T$  predefinido. Dado que el volumen a través de la válvula de control nunca es igual al entregado al paciente, distintas

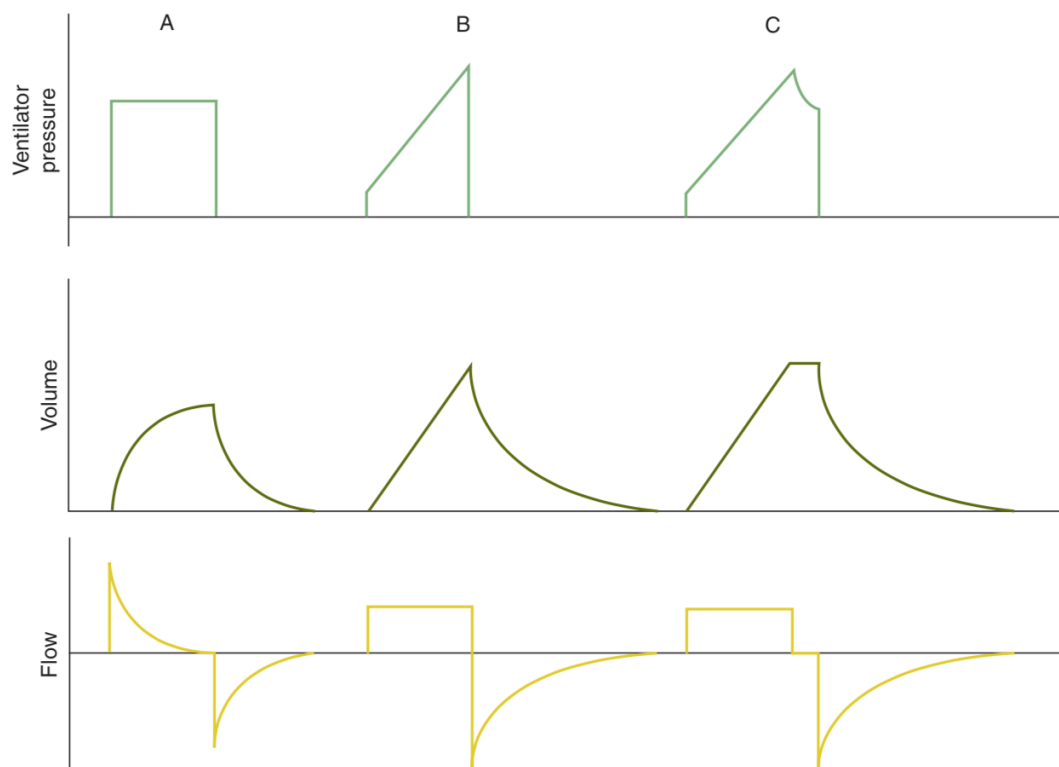


Fig. 2.12: Comparativa entre variables objetivo y de ciclado. A. Objetivo de presión y ciclado por volumen. B. Objetivo de flujo y ciclado por volumen. C. Objetivo de flujo y volumen y ciclado por tiempo. [9]

consideraciones se han de tener en cuenta para garantizar una entrega del volumen deseado, dependiendo del punto del circuito donde se tome la medida.

El uso más habitual del ciclado por flujo es en el modo de presión soporte (PSV). En este caso, la variable de control es la presión, y el ventilador proporciona el flujo necesario para alcanzar cuanto antes el nivel de presión objetivo. En esta dinámica, y obviando el esfuerzo del paciente, el flujo alcanzará rápidamente su máximo y comenzará a disminuir (Fig. 2.12-A); el ciclado de la inspiración se da entonces cuando el valor del flujo disminuye hasta el valor prefijado, que habitualmente es un porcentaje del valor máximo alcanzado en esa respiración (del orden del 25 %). Este umbral, cuando se establece como porcentaje del flujo máximo, se conoce como *expiratory trigger sensitivity* (ETS) [17]; aumentar este valor provoca una disminución del tiempo inspiratorio, y viceversa. Por último, si la variable de ciclado es el tiempo, la espiración comenzará cuando se alcance el tiempo inspiratorio predefinido. En la Fig. 2.12 se ilustra una comparativa entre los términos *objetivo* y *ciclado*.

- **Variable de base.** Este es el parámetro que se controla durante la espiración. Aunque también podrían ser el volumen o el flujo, en la práctica todos los ventiladores modernos utilizan la presión como variable de base. Esta presión se mide y se fija en relación con la presión atmosférica, y no es otra cosa que la PEEP o **PEEP extrínseca**. En la Fig. 2.13 se puede visualizar tanto la PEEP extrínseca como la auto-PEEP, causada por una interrupción prematura de la espiración (véase como el flujo aún no había cesado).

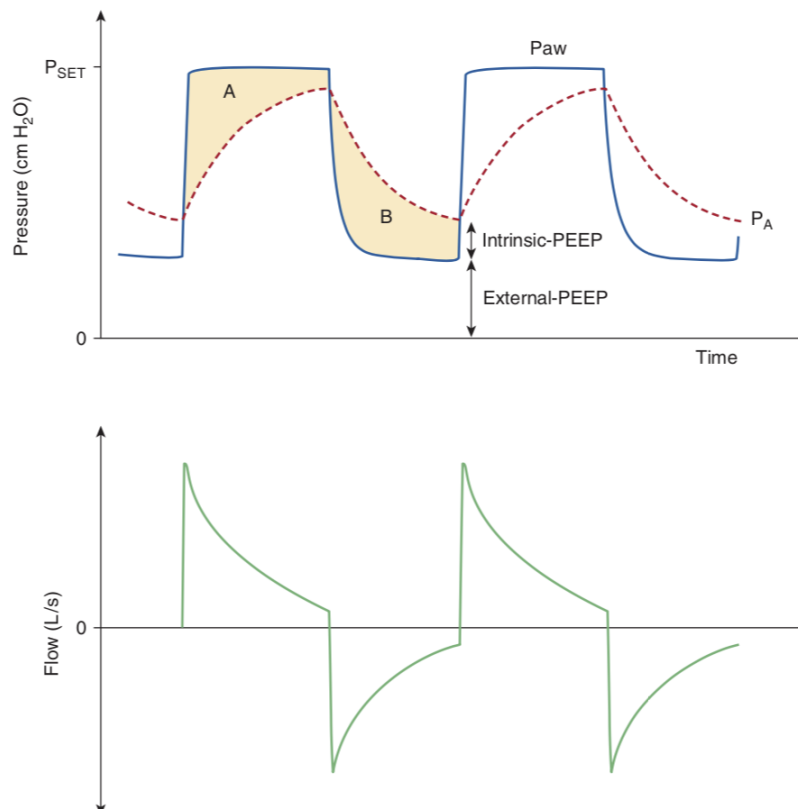


Fig. 2.13: Visualización de la PEEP intrínseca y extrínseca. Presión en la vía aérea ( $P_{aw}$ ), presión alveolar ( $P_A$ ) y flujo durante ventilación controlada por presión.  $A$  y  $B$  representan el producto presión-tiempo disipado, proporcional a las resistencias inspiratoria y espiratoria, respectivamente. [18]

### 2.1.4.2. Modos de ventilación

La ventilación mecánica persigue la seguridad, el confort y la liberación en términos de extubación, y en este proceso los VMs implementan distintos patrones de interacción paciente-ventilador, los modos de ventilación. De acuerdo con [9], los modos se pueden clasificar de forma esquemática según los siguientes parámetros:

- **Variable de control primaria.** Se ha visto que los ventiladores pueden funcionar bien controlando la **presión** o bien el **volumen**, dado que este último es equivalente al control del flujo. Ambos funcionamientos tienen sus ventajas clínicas. Sin entrar en demasiado detalle al respecto, el control de volumen asegura un volumen minuto ( $V_M$ ) —volumen desplazado por la respiración a lo largo de un minuto— más estable, y con ello también los gases en sangre; mientras que los VMs controlados por presión permiten una mejor sincronización con el paciente al no establecer el flujo en un parámetro predefinido.
- **Secuencia respiratoria.** Esto es el patrón de respiraciones mandatorias o espontáneas que el ventilador entrega. Antes de desarrollar los tipos de respiraciones posibles, decir que una respiración se define como un flujo de aire positivo (inspiración) y un flujo de aire negativo (espiración) de igual tamaño pero no necesariamente de igual duración.

Una *respiración espontánea* es, en el contexto de la ventilación mecánica, una respiración disparada y ciclada por el paciente, por lo que es él quien determina tanto el tiempo como el tamaño.

De forma no excluyente, una *respiración asistida* es aquella en la que el ventilador realiza algún tipo de trabajo para el paciente, bien sea aumentando la presión en la vía aérea durante la inspiración o disminuyéndola durante la espiración. Un ejemplo de esto es el modo de presión soporte, en el que el ventilador aumenta la presión en boca hasta el nivel predefinido; pero en este modo las respiraciones también son espontáneas, pues es el paciente quien la dispara y quien la finaliza, bien mediante exhalación activa o pasiva, en cuyo caso es la impedancia del paciente quien marca el punto de terminación de la inspiración. Recordar que en este modo, el ventilador detecta el esfuerzo inspiratorio del paciente bien por flujo o por presión, y cesa el soporte de presión cuando el flujo disminuye hasta el valor prefijado (ETS).

En el caso de la ventilación CPAP, en la que se proporciona una ligera presión constante en las vías respiratorias para mantenerlas abiertas [19], las respiraciones son espontáneas pero no asistidas, pues el ventilador no realiza ningún trabajo para el paciente.

Por último, una *respiración mandatoria* es aquella en la que el paciente no tiene control ni sobre el tiempo ni sobre las dimensiones, y por lo tanto es contraria a la respiración espontánea. Por tanto, es el ventilador quien controla el comienzo y/o el fin de la inspiración.

Con esto en mente, se pueden definir tres posibles secuencias respiratorias:

- ***Continuous spontaneous ventilation (CSV)***. Todas las respiraciones son espontáneas.
- ***Intermittent mandatory ventilation (IMV)***. Se pueden dar respiraciones espontáneas entre las mandatorias. Cuando dichas respiraciones mandatorias son disparadas por el paciente, se suele referir como *synchronized IMV (SIMV)*.
- ***Continuous mandatory ventilation (CMV)***. No se permiten respiraciones espontáneas, pues el objetivo es proporcionar una respiración mandatoria por cada esfuerzo del paciente, a no ser que ya se esté produciendo una o que el paciente no haga esfuerzos por estar paralizado.

Con estas dos variables de control y con las tres secuencias respiratorias comentadas, se puede hacer ya una primera clasificación de los modos de ventilación suficiente para la mayoría de objetivos clínicos, en base a los *cinco patrones de respiración posibles*: dos de control de volumen, VC-CMV y VC-IMV, y tres de control de presión, PC-CMV, PC-IMV y PC-CSV. El patrón VC-CSV no es posible debido a que el control de volumen implica el ciclado por volumen, por lo que la respiración ya no sería espontánea.

A mayores, se puede proporcionar otro nivel de detalle en la clasificación de los modos de ventilación:

- **Esquemas de control del objetivo.** Los *targeting schemes* se refieren a los sistemas de control por realimentación utilizados en los VMs para proporcionar el patrón ventilatorio correspondiente. El esquema genérico de un sistema de control con realimentación es el que se puede observar en la Fig. 2.14, donde el *effector* es el propio ventilador y la planta, el paciente junto con el circuito que lo conecta al VM. El controlador es el elemento que recibe la señal de error —la diferencia entre lo que se quiere obtener y lo que realmente se está consiguiendo— y, mediante distintas operaciones matemáticas, la transforma en la señal que acciona el ventilador y con la dinámica necesaria para alcanzar el objetivo deseado. A partir de este concepto, los modos de ventilación pueden distinguirse también por las siguientes seis formas de control de los objetivos ventilatorios:

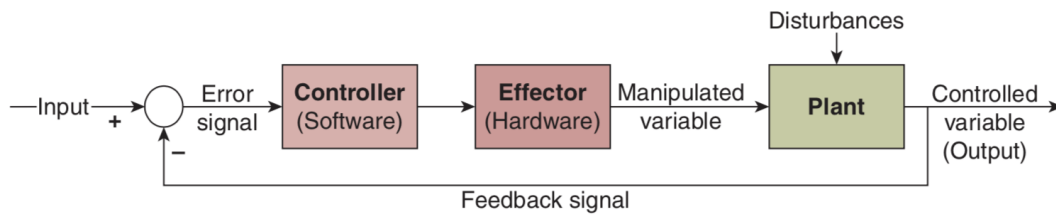


Fig. 2.14: Esquema general de un circuito de control en lazo cerrado [9]

- **Consigna.** Es el esquema más simple, donde el operador especifica unos valores objetivo tales como volumen tidal y flujo inspiratorio en modos de control de volumen, o presión inspiratoria y tiempo inspiratorio en control de presión, y el ventilador intentará entregarlos al paciente. Otros parámetros generales pueden ser la frecuencia respiratoria, la PEEP o la concentración de oxígeno.
- **Dual.** Tal como se ha explicado anteriormente, teniendo en cuenta la ecuación del movimiento (2.2) solo una variable puede ser la independiente en un determinado momento (presión o volumen), pero es posible que el ventilador cambie entre las dos durante una misma inspiración, obteniendo un esquema de consigna dual. Esto se puede conseguir de dos maneras:
  - Comenzar la inspiración en control de volumen y cambiar a presión si uno o más valores predefinidos es alcanzado, por ejemplo una presión máxima en la vía aérea deseada.
  - Comenzar la inspiración en control de presión y cambiar a volumen si, por ejemplo, no se ha alcanzado el volumen tidal prefijado cuando el flujo decae hasta un cierto valor.

Este esquema de control proporciona la seguridad de garantizar un volumen tidal mínimo a la vez que el confort para el paciente derivado de la sincronía del flujo con el control de presión.

- **Servo.** El término *servo control* se aplica aquí a los esquemas de control en los que la salida del ventilador sigue automáticamente una entrada cambiante, haciendo que el trabajo del ventilador se ajuste a la demanda del paciente con más fidelidad.

Un ejemplo de esto es el modo PAV (*Proportional-Assist Ventilation*), en el que la salida del ventilador sigue y amplifica señales dependientes del esfuerzo del paciente, como el volumen y el flujo.

- **Adaptativo.** En este caso, el ventilador es capaz de ajustar algunos de los parámetros de forma que se consigan los objetivos deseados, teniendo en cuenta que las condiciones del paciente son cambiantes. Los siguientes modos implementan las cuatro maneras en las que se puede aplicar el esquema de control adaptativo básico:
  - *Control de volumen regulado por presión.* La presión inspiratoria se ajusta automáticamente para conseguir el objetivo de volumen tidal medio.

- *Ritmo de ventilación mandatorio.* La presión inspiratoria se ajusta automáticamente para mantener el objetivo de frecuencia respiratoria espontánea.
  - *Flujo adaptativo/tiempo-i adaptativo.* Tiempo y flujo inspiratorios ajustados automáticamente para mantener el objetivo de ratio tiempo inspiratorio-tiempo espiratorio en 1:2.
  - *Ventilación minuto mandatoria.* Frecuencia respiratoria ajustada automáticamente para mantener el objetivo de volumen minuto.
- **Óptimo.** Este esquema es una configuración avanzada del adaptativo. El VM ajusta automáticamente los objetivos de la ventilación de forma que se minimice o maximice alguna característica general del rendimiento. El único modo de este tipo disponible comercialmente es el *Adaptive-support ventilation (ASV)*, en el que a partir de tan solo el peso del paciente y de un modelo matemático, se ajustan los objetivos de ventilación de forma que se minimice el trabajo respiratorio, se imite la respiración natural, se estimule la respiración espontánea y se reduzca el tiempo de destete o extubación. En la Fig 2.15 se puede observar el esquema de control de este modo de ventilación.

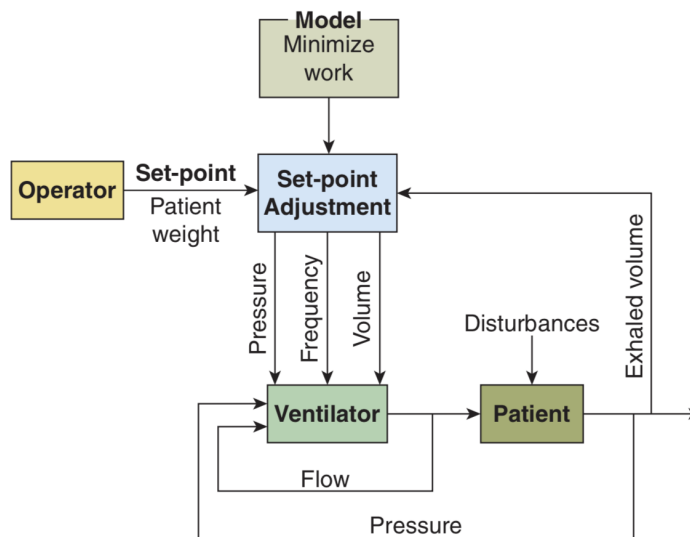


Fig. 2.15: Esquema óptimo de control del objetivo, del modo de ventilación ASV [9]

- **Inteligente.** Un paso más allá en el esquema de control adaptativo, caracterizado por el uso de técnicas de inteligencia artificial. Como ejemplo de esto está el modo SmartCare/PS de Dräger®, una variación del modo de presión soporte basado en un *sistema experto de reglas* para conseguir una extubación automática del paciente. Este proceso se basa en la monitorización del paciente y, a partir de dichas reglas, aplicar una ventilación cada vez menos agresiva, hasta mostrar un mensaje aconsejando al clínico el destete.

Si bien este modo de inteligencia artificial reduce el tiempo de permanencia bajo ventilación mecánica, sus ventajas no son tan evidentes en un entorno donde el personal clínico disponga de un amplio conocimiento *natural* al respecto, pues las reglas que lo rigen también han sido elaboradas por expertos. Más prometedor resulta el uso de las *redes neuronales*, pues estas, frente a la rigidez de los modelos matemáticos y los sistemas de reglas, son capaces de aprender tal y como lo hacen nuestras propias neuronas.

A modo de resumen, los modos de ventilación pueden clasificarse de acuerdo a su variable de control, su secuencia respiratoria y su esquema de control de objetivo. En el caso de IMV, un segundo esquema de control aplicaría a las respiraciones secundarias. En las Fig. 2.16 y 2.17 se



muestran distintos algoritmos para la determinación tanto de la variable de control como de la secuencia respiratoria de un modo.

### 2.1.4.3. Modos de interés

A continuación, se detalla el funcionamiento de los modos de ventilación que son de más interés para el presente proyecto:

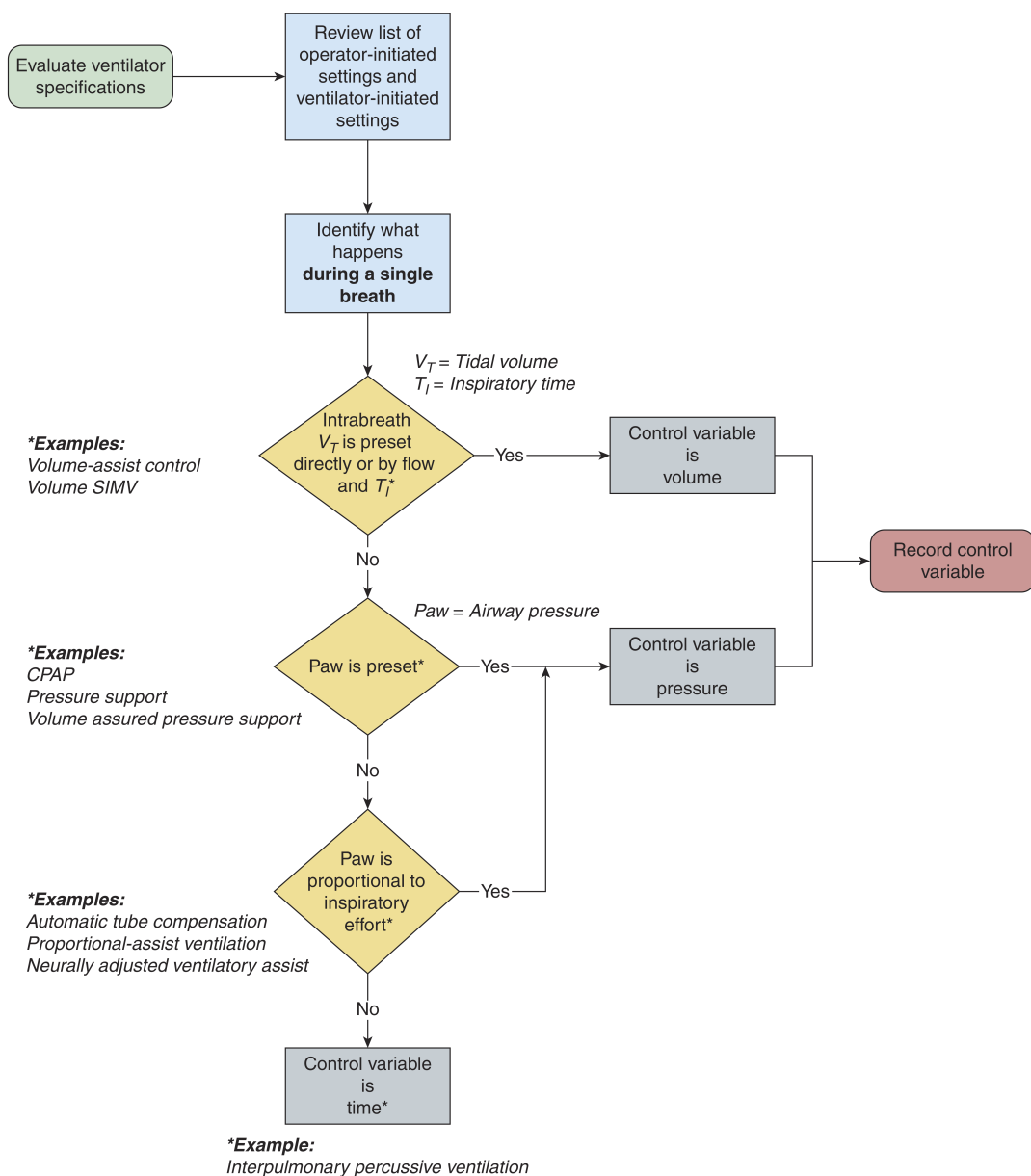


Fig. 2.16: Algoritmo para determinar la variable de control de un modo ventilatorio [9]

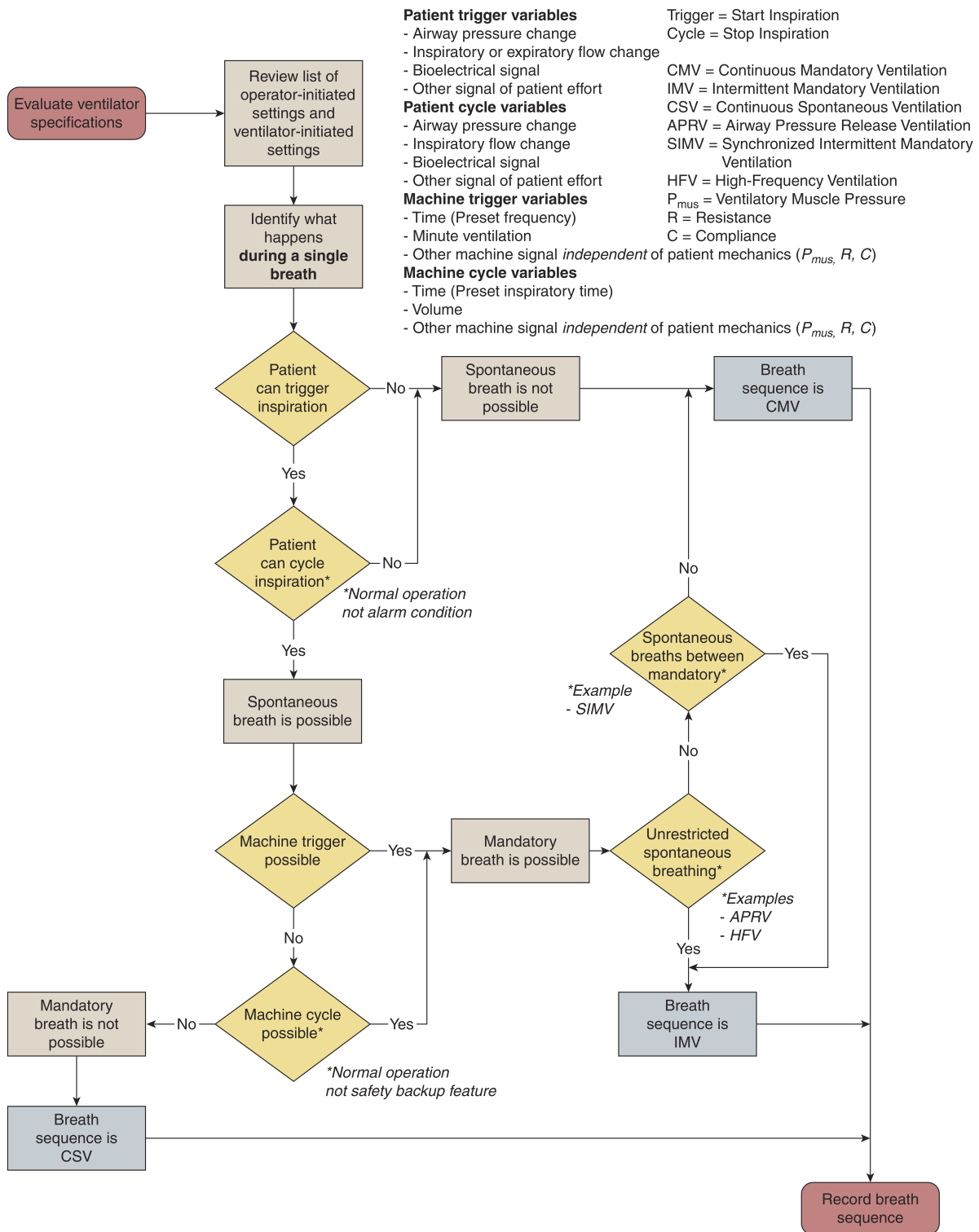


Fig. 2.17: Algoritmo para determinar la secuencia respiratoria de un modo ventilatorio [9]

## VC-CMV

A veces escrito como simplemente CMV, este es un modo controlado por volumen y donde todas las respiraciones son mandatorias.

Por un lado, esto quiere decir que la variable a controlar es el volumen, normalmente a través del flujo, y por tanto la presión es la variable dependiente. Un aumento en la impedancia de la *planta* (sistema respiratorio más circuito del paciente) resultaría en una mayor presión de la vía aérea [20].

Por el otro lado, la secuencia respiratoria es CMV porque, independientemente de la variable de disparo, el ciclado es por tiempo [11].

Los parámetros principales a ajustar en este modo son el  $V_T$ , la FR y la forma de suministrar el  $V_T$ , además de otros parámetros generales como la PEEP o las alarmas. En la manera de suministrar el volumen tidal intervienen el valor máximo del flujo inspiratorio, su forma de onda (normalmente constante o rampa descendiente) y el tiempo inspiratorio ( $T_i$ ). Sin embargo, no todos estos parámetros son ajustables, pues tal como se muestra en la ecuación (2.6), el  $V_T$ , el  $T_i$  y el flujo inspiratorio ( $\dot{V}_i$ ) son mutuamente interdependientes [11]. Por tanto, existen ventiladores en los que el  $T_i$  es calculado matemáticamente a partir del  $V_T$  y del  $\dot{V}_i$ , y otros en los que, partiendo de la forma de onda del flujo, el  $T_i$  y el  $V_T$ , se calcula el valor máximo del flujo.

$$\dot{V}_i = \frac{V_T}{T_i} \quad (2.6)$$

Para un flujo del tipo constante o escalón, la forma de onda obtenida sería como la que se puede observar en la Fig. 2.12-B. Podría deducirse, entonces, que en realidad este modo es ciclado por volumen; sin embargo, aun cuando se ajusta el  $V_T$  y el flujo inspiratorio en forma y magnitud, se está determinando el tiempo que durará la inspiración, por lo que el ciclado es por tiempo [20].

## PC-CMV

Al igual que en el anterior, en este modo las respiraciones son siempre mandatorias y cicladas por tiempo. Por contra, la variable a controlar es la presión, por lo que tanto el flujo como el volumen inspirados serán función de la resistencia y compliancia del sistema [11].

Por consiguiente, se debe ajustar un nivel de presión de control ( $P_{control}$ ) que será constante a lo largo de toda la inspiración. De este modo, el  $V_T$  se verá disminuido ante una menor compliancia o una mayor resistencia de la vía aérea, y viceversa.

Dado que la variable de ciclado es el tiempo, será necesario ajustar también el tiempo inspiratorio, así como la FR y los demás parámetros comunes a otros modos. A mayores, casi todos los ventiladores que implementan este modo cuentan con un parámetro adicional que es el *Rise Time* o tiempo de subida. Este parámetro establece el tiempo que tarda en alcanzar la salida del ventilador la presión de control, desde el nivel de presión de base y desde que comienza la inspiración, de forma lineal (véase Fig. 2.18) [20].

## PSV

El modo de presión soporte o *Pressure Support Ventilation* proporciona asistencia a las respiraciones espontáneas del paciente (CSV) mediante el control de la presión.

Similar al modo PC-CMV en cuanto a que comparte los parámetros  $P_{control}$  y  $T_e$ , pero con la diferencia de que el ciclado es por flujo. Ante la consigna de una presión constante, el flujo (dependiente del esfuerzo e impedancia del paciente) alcanzará rápidamente su pico máximo y comenzará a decaer. El umbral de

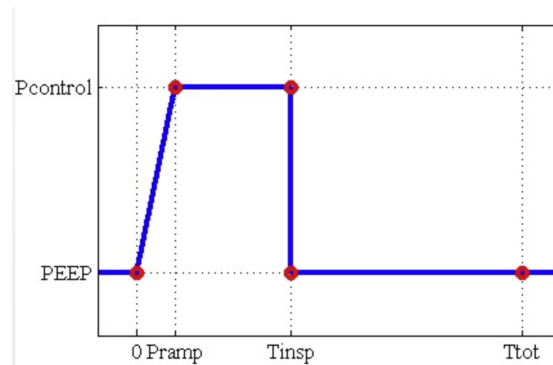


Fig. 2.18: Forma de onda de la presión generada por el ventilador en el modo PC-CMV [11]

flujo para comenzar la espiración viene determinado por la ETS como una fracción de ese flujo máximo (véase Fig. 2.19).

Las respiraciones son todas espontáneas puesto que son disparadas por el esfuerzo del paciente y, además, el flujo que las cicla es función de este esfuerzo y de las características instantáneas del paciente.

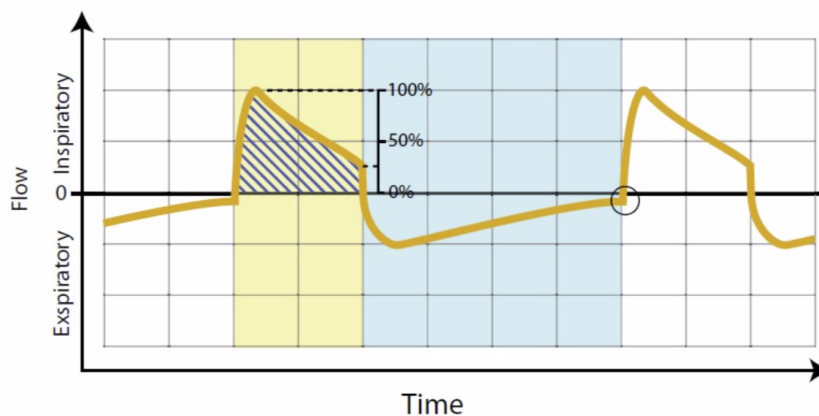


Fig. 2.19: Curva de flujo en el modo PSV con una ETS del 25% [11]

## PAV

El modo *Proportional Assist Ventilation* es una forma de ventilación mecánica menos convencional, que utiliza un esquema de control del tipo servo. El objetivo de este modo es normalizar la relación entre el esfuerzo instantáneo del paciente y las consecuencias ventilatorias (flujo y volumen entregados); o dicho de otro modo, suministrar una ventilación proporcional al esfuerzo inspiratorio del paciente en cada momento [21].

En la Fig. 2.20 se puede observar cómo influyen distintos modos de ventilación en la relación entre el esfuerzo del paciente y, por un lado, la presión entregada, y por el otro, las consecuencias ventilatorias (flujo y volumen). Mientras que en los demás modos de ventilación la relación esfuerzo-consecuencias ventilatorias solo es normal en un punto (donde se intersecan la recta  $N$  y la correspondiente a cada modo), el modo PAV persigue normalizar esa relación a todos los niveles de esfuerzo del paciente.

Esto lo consigue a base de no predeterminedar ni la  $P_{vent}$ , ni el flujo ( $\dot{V}$ ) ni el volumen ( $V$ ). La  $P_{vent}$  es, por tanto, función tanto de  $\dot{V}$  como de  $V$ , ambos determinados por el esfuerzo del paciente. Reformulando la ecuación del movimiento (2.2), esta quedaría tal que:

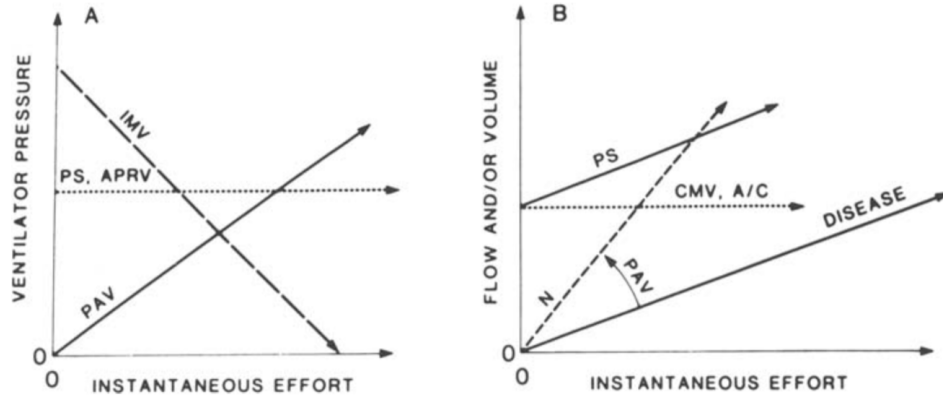


Fig. 2.20: Efecto de distintos modos de ventilación en la relación entre esfuerzo instantáneo del paciente y presión entregada(A)/consecuencias ventilatorias(B) ( $N$  = normal;  $DISEASE$  = afección;  $IMV$  = respiración ciclada por volumen;  $CMV$  y  $A/C$  (*assist/control*) son cicladas por volumen (de forma indirecta si el  $\dot{V}_i$  es función del  $V_T$  y del  $T_i$ );  $PS$  = *pressure support*;  $APRV$  = *airway pressure release ventilation*;  $PAV$  = *proportional assist ventilation*) [21]

$$P_{mus}(t) + [V(t)K_1 + \dot{V}(t)K_2] = V(t)E + \dot{V}(t)R \quad (2.7)$$

donde  $K_1$  es la relación entre  $P_{vent}$  y el volumen inspirado (en unidades de elastancia,  $\text{cmH}_2\text{O}/\text{L}$ ) y  $K_2$  entre  $P_{vent}$  y el flujo inspirado (en unidades de resistencia,  $\text{cmH}_2\text{O}/(\text{L}/\text{s})$ ).

Con este modo, un aumento de la  $P_{mus}$  provocaría un incremento en la presión total aplicada y, con ello, en el flujo y en el volumen inspirados. Esto a su vez causaría un aumento en la  $P_{vent}$  aplicada, provocando nuevamente un que  $\dot{V}$  y  $V$  aumentasen. Esto es precisamente la definición de un sistema con realimentación positiva, inestable por naturaleza.

Sin embargo, esta realimentación positiva se ve compensada por la propia realimentación negativa de las propiedades mecánicas del paciente (resistencia y elastancia). En la Fig. 2.21 se ilustra tanto un esquema del modelo de ventilador y control del modo PAV, donde se visualiza la realimentación del flujo y volumen, como un ejemplo para demostrar la estabilidad del sistema. En este ejemplo, se consideran dos hipótesis para facilitar el entendimiento de la relación entre el sistema y el paciente: que los cambios en las variables se dan todos de forma discreta y que la presión total se aplica únicamente contra la elastancia del paciente ( $R = 0$ ). Esta elastancia se expresa en unidades arbitrarias y con valor de 1 unidad de presión/unidad de volumen.

Partiendo de un estado inicial estable donde la presión total es nula, el paciente realiza un esfuerzo provocando un cambio en la  $P_{mus}$  de 4 unidades de presión. De acuerdo con la elastancia, se mueven 4 unidades de volumen del ventilador al paciente. En el siguiente instante, la  $P_{vent}$  ya es calculada en base al valor medido de volumen y la elastancia. Suponiendo un valor de  $K_1$  de 0,5 unidades de presión/unidad de volumen, la  $P_{vent}$  aplicada será de 2 unidades de presión. El valor de presión total es ahora de 6 unidades, causando nuevamente un incremento en el volumen, esta vez, de 2 unidades. Esto, a su vez, vuelve a causar un incremento en la  $P_{vent}$ , de 1 unidad de presión. Se puede ver como, al ser la ganancia menor que el valor de la elastancia, el sistema tiende a la estabilidad, sin importar la magnitud del cambio en la  $P_{mus}$  (ver el segundo escalón en la Fig. 2.21)

Dado que el retroceso elástico en cualquier instante es mantenido por ambas la  $P_{mus}$  y la  $P_{vent}$ , si el paciente reduce su contribución a la presión total (flecha en Fig. 2.21) esta no será capaz de mantener ese retroceso de los pulmones y comenzará la espiración.

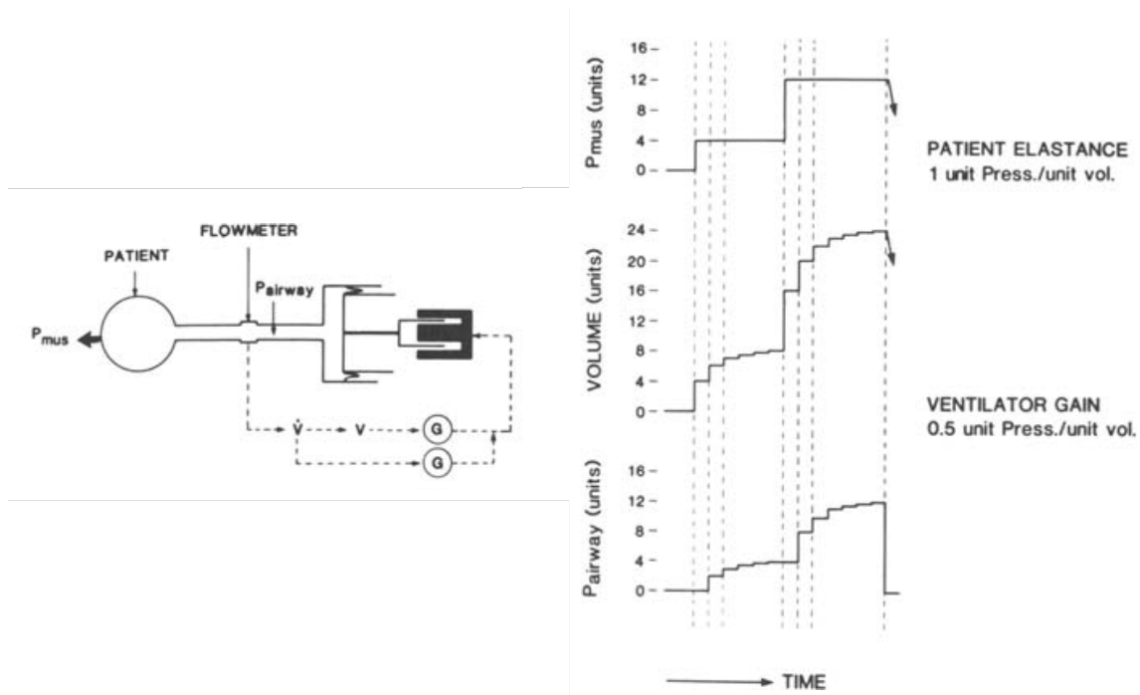


Fig. 2.21: Diagrama y funcionamiento del modo PAV ( $P_{airway}$  = presión generada por el ventilador ( $P_{vent}$ );  $G$  = ganancias de flujo y de volumen ( $K_1$  y  $K_2$ )) [21]

Se puede deducir que el valor de amplificación de la componente elástica en  $P_{mus}$  es función de la ganancia de volumen ( $K_1$ ) y de la propia elastancia del paciente. Puesto que el total de presión elástica ( $V \cdot E$ ) es igual a la suma de las componentes elásticas de  $P_{mus}$  y de  $P_{vent}$ , se puede deducir que:

$$\begin{aligned} P_{mus}(el) &= V \cdot E - P_{vent}(el) \\ &= V \cdot E - V \cdot K_1 \\ &= V(E - K_1) \end{aligned} \quad (2.8)$$

De (2.8) se obtiene que el factor de amplificación elástica ( $F(el)$ ), definido como la relación entre la presión elástica total y la desarrollada por el paciente, es:

$$\begin{aligned} F(el) &= \frac{V \cdot E}{V(E - K_1)} \\ &= \frac{E}{E - K_1} \end{aligned} \quad (2.9)$$

Se tiene entonces que, para un caso como el anterior donde la  $K_1$  es la mitad de la elastancia, el factor de amplificación es de 2. En caso de que la  $K_1$  fuese igual o superior a  $E$ , la amplificación sería infinita y el sistema podría volverse inestable. Análogamente, la amplificación de la componente resistiva de  $P_{mus}$  viene dada por:

$$F(res) = \frac{R}{R - K_2} \quad (2.10)$$

Podría darse el caso de dar soporte ventilatorio teniendo en cuenta solo una de las dos componentes, resistiva o elástica. En el primer caso, y dado que el flujo alcanza su máximo al principio de la inspiración

y luego decae, el ventilador amplificaría el esfuerzo solo en este intervalo, dejando al paciente sin soporte hacia el final de la inspiración, cuando el esfuerzo es mayor. Del mismo modo, ventilando solo en relación a la componente elástica privaría al paciente de soporte al inicio de la inspiración. Es por esto que, a fin de normalizar la relación esfuerzo-consecuencias ventilatorias, se debe proporcionar soporte en función de ambas componentes. Esto puede hacerse de cuatro formas diferentes:

- **Soporte en proporción a la  $P_{mus}$  estimada.** La  $P_{mus}$  instantánea podría estimarse a partir de la ecuación del movimiento (2.2), y con esto calcular la  $P_{vent}$  a entregar. Sin embargo, es fácil deducir matemáticamente como, si se seleccionan las mismas  $F(el)$  y  $F(res)$ , el efecto sería el mismo sin tener que estimar la  $P_{mus}$ . La  $P_{mus}$  total se vería entonces amplificada por el mismo factor independientemente del peso de cada componente, y por tanto también del tiempo durante la inspiración. Esta estrategia es ideal para casos en los que el problema está en la habilidad de generar presión y las propiedades mecánicas son relativamente normales.
- **Ajuste de  $K_1$  y  $K_2$  de acuerdo con la naturaleza de la anomalía mecánica.** En este caso, las ganancias se ajustarían según la diferencia entre los valores de resistencia y elastancia del paciente y los valores normales. De este modo, el ventilador afrontaría el trabajo extra producido por la afección mientras que el paciente asumiría una carga normal. Este método normaliza la relación entre la  $P_{mus}$  y las consecuencias ventilatorias, pero no así entre la  $P_{mus}$  y el esfuerzo, por lo que es apropiado para pacientes con anomalías mecánicas pero un aparato neuromuscular normal.
- **Combinación de las anteriores.** Cuando la afección es mixta —debilidad neuromuscular y anomalías mecánicas— las ganancias de flujo y volumen pueden ser cada una la suma de dos componentes: una para compensar las anomalías mecánicas y otra no específica para la amplificación de la  $P_{mus}$ .

A modo de ejemplo, considerar un paciente con una elastancia de 40 cmH<sub>2</sub>O/L y una resistencia de 6 cmH<sub>2</sub>O/(L/s), en lugar de unos valores normales de 15 cmH<sub>2</sub>O/L y 4 cmH<sub>2</sub>O/(L/s). Entonces,  $K_1$  podría ser 32,5 cmH<sub>2</sub>O/L —25 para compensar la elevada elastancia y 7,5 para reducir la elastancia normal a la mitad— y  $K_2$  podría ser 4 cmH<sub>2</sub>O/(L/s) —2 para compensar la elevada resistencia y otras 2 unidades para reducir la resistencia normal a la mitad—. En este caso, tanto la relación  $P_{mus}$ -consecuencias ventilatorias como la relación  $P_{mus}$ -esfuerzo del paciente están normalizadas.

- **Ajuste de  $K_1$  y  $K_2$  de acuerdo con el confort del paciente.** Las tres anteriores modalidades requieren conocer con cierta exactitud los valores de resistencia y elastancia del paciente, lo cual no siempre es posible. En este caso, se podría utilizar los precisos mecanismos sensoriales del propio paciente, para ajustar las ganancias a base de prueba y error de forma que resulten en la mayor comodidad para el paciente.

En cuanto al funcionamiento de este modo, en la Fig. 2.22 puede verse una representación del ventilador utilizado en el ensayo clínico llevado a cabo por Younes et al. [22].

En este ventilador, la válvula espiratoria ( $EV$ ) está cerrada durante la inspiración mediante la conexión a la cámara de presión a través de una electroválvula ( $SV_2$ ), la cual es activada con un umbral de flujo de 0,05 L/s. La señal de flujo medida con el neumotacógrafo es integrada para obtener el volumen inspirado, y esta señal de volumen es reseteada a cero al final de cada inspiración (cuando el flujo cae por debajo de 0,05 L/s). Estas señales de flujo y volumen son las utilizadas para accionar el motor y proporcionar la  $P_{vent}$ .

Al final de la inspiración,  $SV_2$  es ventilada a la habitación, causando que la válvula espiratoria se abra y la exhalación comience. Al mismo tiempo, el pistón es retraído a su posición inicial y la cámara se renueva con gas fresco, bien desde la habitación a través de la válvula de un sentido con umbral de presión  $V_2$ , o bien de una fuente de gas a través de la electroválvula  $SV_1$ . Una vez el pistón ha vuelto a su posición inicial, la cámara es presurizada a un nivel de PEEP configurable.

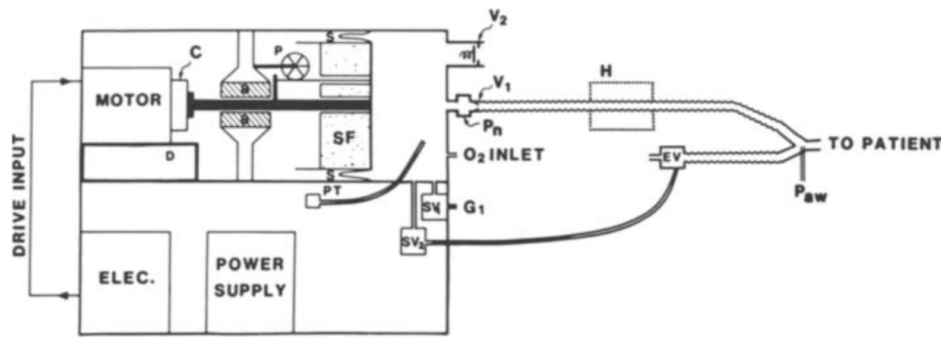


Fig. 2.22: Representación esquemática de una unidad PAV móvil ( $C$  = bobina;  $EV$  = válvula espiratoria;  $G$  = entrada de gas;  $H$  = humidificador;  $P$  = potenciómetro;  $P_{aw}$  = sensor de presión de las vías respiratorias;  $P_n$  = neumotacógrafo;  $PT$  = transductor de presión;  $S$  = junta;  $SF$  = refuerzo de poliestireno;  $V_1$  = válvula antirretorno;  $V_2$  = válvula de resorte) [22]

Con esta configuración, el sistema solo detecta el esfuerzo inspiratorio cuando este es suficiente para disminuir la presión en la vía aérea ( $P_{aw}$ ) por debajo de la PEEP extrínseca, o de la PEEP total en presencia también de auto-PEEP.

## 2.2. ENTORNO MATLAB

La plataforma MATLAB®, herramienta de trabajo en este proyecto, es uno de los entornos de programación y cálculo numérico más extendidos en el mundo de la ingeniería y la ciencia [23]. Permite, entre otras cosas, el desarrollo de algoritmos, la creación de aplicaciones gráficas, el modelado de sistemas e incluso la generación e implementación de código en dispositivos *hardware*.

Las aplicaciones de MATLAB son innumerables; por citar algunas, este entorno tiene utilidad en áreas como la ingeniería de control, la inteligencia artificial, el procesamiento de señales, la biología computacional, la robótica o los sistemas embebidos. Para esto, MATLAB combina un entorno de escritorio hecho a medida para el análisis iterativo y los procesos de diseño con un lenguaje de programación pensado expresamente para las matemáticas.

### 2.2.1. Simulink/Simscape

Dentro de MATLAB se integra el *software* Simulink®, un entorno gráfico ideado para el modelado y simulación de sistemas de todo tipo. Este, además, y al igual que MATLAB, se completa con cientos de extensiones creadas específicamente para tareas más concretas (p.ej. visión artificial o análisis de datos).

La ventaja que ofrece Simulink es que permite el diseño de modelos mediante el uso de componentes en diagramas de bloques, sin necesidad de escribir código adicional, lo cual permite modelar sistemas complejos de forma muy sencilla.

Además, los sistemas pueden modelizarse en diferentes dominios físicos, gracias al uso de las librerías de Simscape™. Este producto integra en Simulink componentes que modelan sistemas físicos de dominios como el eléctrico, mecánico, térmico o de aire húmedo, entre otros. Además, pueden crearse otros componentes definiendo su comportamiento mediante un lenguaje de modelado basado en ecuaciones, con sintaxis heredada de MATLAB.



### 2.2.1.1. Modelo de ventilador mecánico y paciente

De la integración de estos tres productos —MATLAB, Simulink y Simscape—, surge el modelo desarrollado por The MathWorks, Inc. que se ha tomado como punto de partida en la realización de este trabajo. Este ejemplo (véase Fig. 2.23), modela un sistema formado por un ventilador mecánico de presión positiva conectado al sistema respiratorio de un paciente [24].

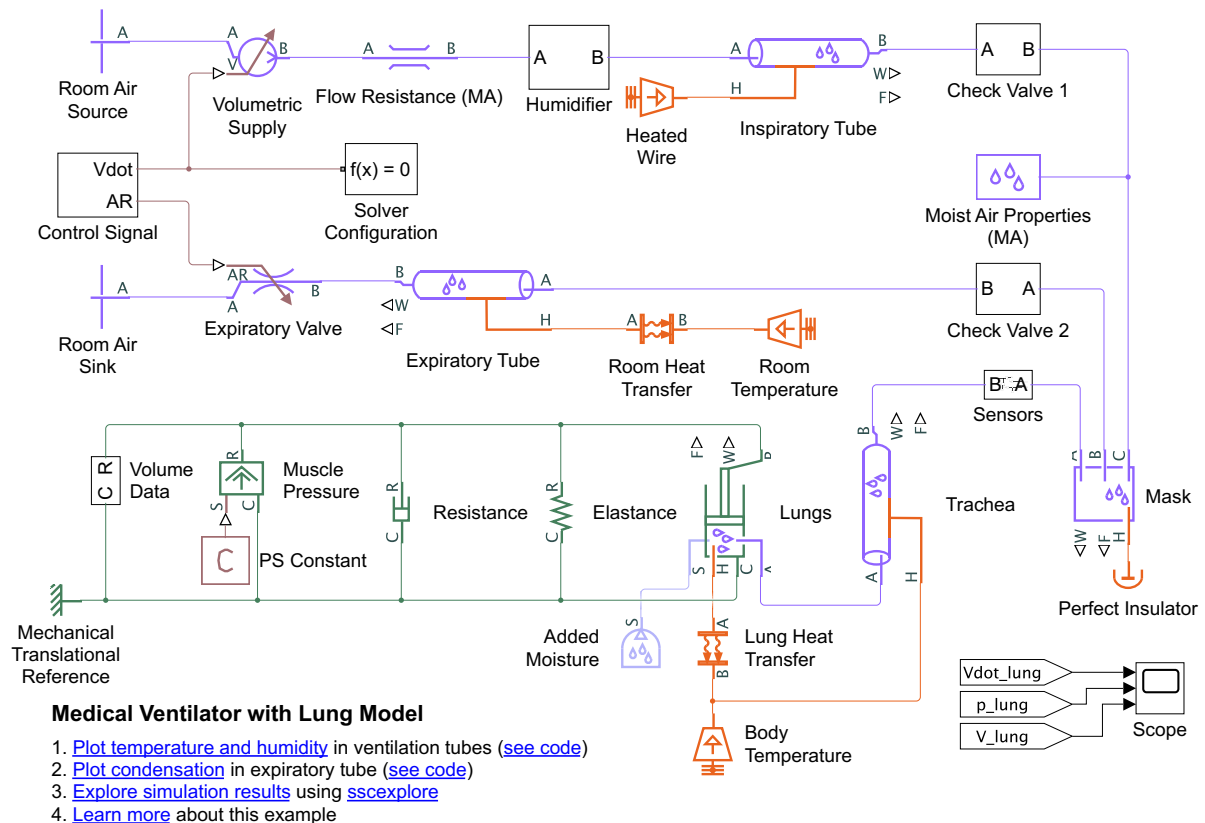


Fig. 2.23: Modelo de partida en Simulink/Simscape de un ventilador mecánico de presión positiva y un sistema respiratorio [24]

Se puede observar como este modelo incorpora elementos de distintos dominios; en concreto, se incluyen componentes de aire húmedo (de color violeta), térmicos (de color naranja) y mecánicos (de color verde). El color marrón se corresponde con elementos comunes a la librería de Simscape, y sus señales llevan asociadas unas unidades físicas concretas. Los elementos en negro se corresponden con bloques, señales y subsistemas del propio entorno Simulink.

A continuación, se enumeran y comentan los distintos elementos incluidos en el modelo:

- **Volumetric Supply.** Dispositivo que proporciona un flujo de aire húmedo del valor de la señal  $V$  de entrada.
- **Room Air Source y Room Air Sink.** Respectivamente, son los puntos de entrada y salida del aire en el circuito.
- **Control Signal.** Subsistema que controla las señales de flujo al dispositivo de suministro de aire y de apertura a la válvula espiratoria. En la Fig. 2.24 se puede ver que ambas señales son cuadradas y opuestas, con sus respectivas ganancias para ajustarse a la magnitud adecuada y con funciones de transferencia que introducen sendos retrasos (para suavizar los cambios). Los bloques *Simulink-PS*

*Converter* transforman la señal de Simulink en una señal física, en este caso con unidades de L/s y m<sup>2</sup>, respectivamente.

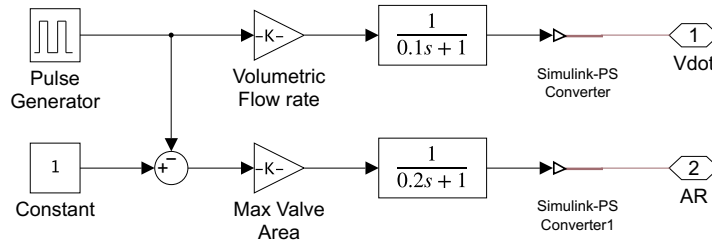


Fig. 2.24: Detalle del subsistema *Control Signal* en el modelo de partida [24]

- **Solver Configuration.** Bloque que establece los ajustes del *solver* para la simulación.
- **Flow Resistance (MA).** Elemento que representa una caída de presión en una red de aire húmedo.
- **Humidifier.** Subsistema que modela un humidificador de aire en la rama inspiratoria del circuito del paciente (véase Fig. 2.25).

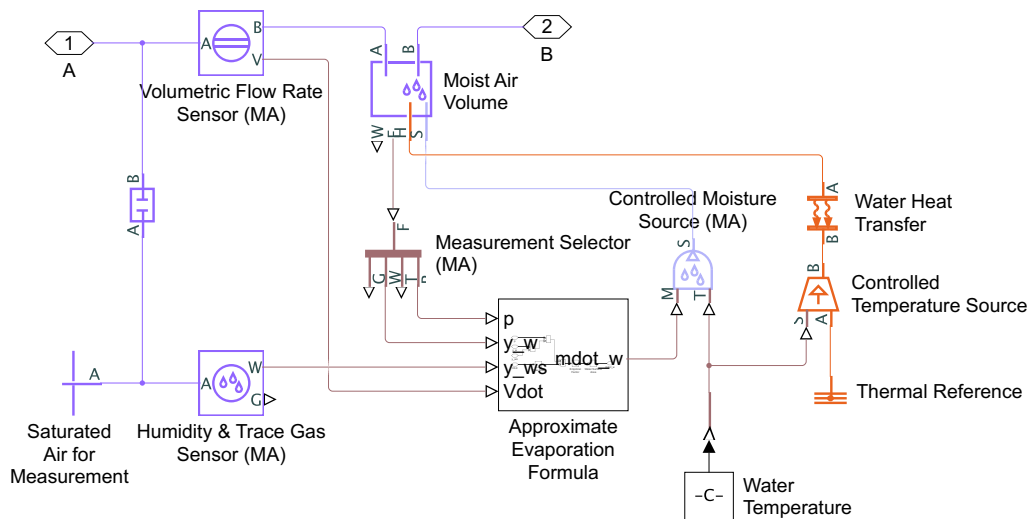


Fig. 2.25: Detalle del subsistema *Humidifier* en el modelo de partida [24]

- **Inspiratory Tube.** Bloque que modela la dinámica de un flujo de aire húmedo en una tubería, teniendo en cuenta las pérdidas por fricción viscosa y la transferencia de calor por convección con las paredes.
- **Heated Wire.** Fuente de energía en una red térmica que, en este caso, representa el cable calefactor que calienta las paredes del tubo inspiratorio.
- **Check Valve.** Subsistema que modela una válvula antirretorno (véase Fig. 2.26). Nótese que tanto la *Check Valve 1* como la *Check Valve 2* son idénticas, pero con sentido opuesto.
- **Moist Air Properties (MA).** Bloque que establece las propiedades del aire húmedo en la red.

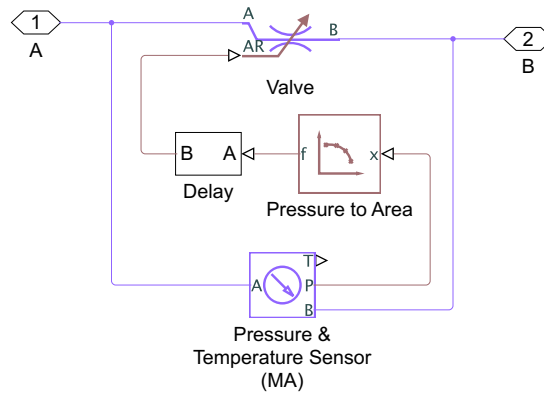


Fig. 2.26: Detalle del subsistema *Check Valve* en el modelo de partida [24]

- **Mask.** Bloque que simula la máscara que actúa de interfaz entre el circuito y el paciente, modelando el almacenamiento de masa y energía en una red de aire húmedo. El uso conjunto con el bloque *Perfect Insulator* modela el aislamiento térmico (perfecto) de la máscara.
- **Expiratory Tube.** Del mismo modo que el *Inspiratory Tube*, representa la rama espiratoria del circuito del paciente. Junto con los elementos *Room Heat Transfer* y *Room Temperature*, se modela la disipación de calor por convección con la temperatura ambiente.
- **Expiratory Valve.** Componente que modela la válvula espiratoria, con una pérdida de presión función de la restricción de área, indicada mediante la señal generada por el bloque *Control Signal*.
- **Sensors.** Subsistema que implementa los sensores de flujo y de presión (véase Fig. 2.27).

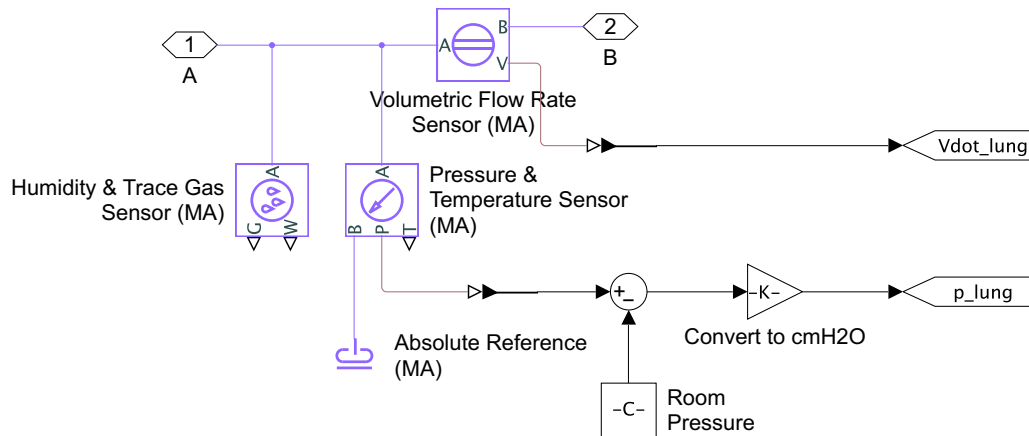


Fig. 2.27: Detalle del subsistema *Sensors* en el modelo de partida [24]

- **Trachea.** Mismo elemento que *Inspiratory Tube* y *Expiratory Tube*, esta vez modelando la tráquea, con una temperatura de las paredes igual a la temperatura del paciente.
- **Lungs.** Bloque que modela los pulmones como una interfaz entre una red de aire húmedo y una red mecánica traslacional. En este proceso, se tiene en cuenta el volumen de aire húmedo, la presión y temperatura en función de la compresibilidad y capacidad termal de este volumen, y la condensación del agua líquida cuando se alcanza la saturación. Del mismo modo que el *Expiratory*

*Tube*, se modela la transferencia de calor con el cuerpo debido al movimiento del aire en los pulmones.

- **Elastance.** Resorte mecánico lineal ideal que representa la elastancia del sistema respiratorio.
- **Resistance.** Amortiguador mecánico viscoso de traslación ideal que representa la resistencia del sistema respiratorio.
- **Muscle Pressure.** Fuente ideal que genera una fuerza mecánica proporcional a la señal física de entrada. Dicha señal de entrada viene modelada por el bloque *PS Constant*, que modela la presión muscular como una constante.
- **Volume Data.** Subsistema que modela un sensor del volumen de los pulmones a partir del desplazamiento de la red mecánica traslacional que los modela (véase Fig. 2.28).

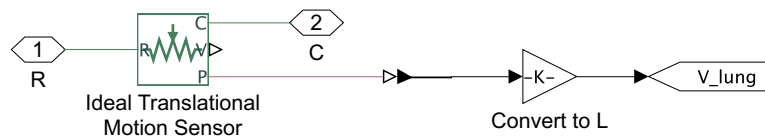


Fig. 2.28: Detalle del subsistema *Volume Data* en el modelo de partida [24]

- **Mechanical Translational Reference.** Elemento que representa un punto de referencia en la red mecánica traslacional, análogamente a una tierra en una red eléctrica.
- **Vdot\_lung, p\_lung y V\_lung.** Los elementos del nivel superior del modelo identificados con estos nombres son del tipo *From*. El nombre visible en estos elementos corresponde al de una etiqueta, la cual se asocia con una señal concreta del modelo mediante el elemento hermano *Goto* (véanse Fig. 2.27 y 2.28). El parámetro de visibilidad de la etiqueta define el ámbito de accesibilidad de la señal (local, global o definido mediante un elemento del tipo *Goto Tag Visibility*).
- **Scope.** Bloque de Simulink que visualiza las señales generadas durante la simulación, en este caso las medidas por los sensores y asociadas a las etiquetas (*Vdot\_lung*, *p\_lung* y *V\_lung*).

### 2.2.2. App Designer

App Designer es el *software* de MATLAB ideado para la creación de *apps*. Dispone de un entorno interactivo para el desarrollo tanto de la interfaz gráfica de usuario (GUI) como del comportamiento de la aplicación [23].

Mediante el uso del lienzo o *Design View* se pueden agregar componentes visuales al *front-end* de la aplicación a base de arrastrar y colocar. Algunos ejemplos de estos componentes son: botones, etiquetas, campos numéricos, gráficos, paneles, potenciómetros o lámparas (véase Fig. 2.29). El entorno genera automáticamente código orientado a objetos que detalla la distribución y el diseño de la aplicación. Desde la vista *Code View*, que integra una versión del editor de MATLAB, se puede definir el *back-end* de la *app*, indicando el comportamiento de sus componentes ante interacciones del usuario.

Dado que el lenguaje utilizado para modelar el comportamiento de la aplicación es orientado a objetos, la *app* se define como una nueva clase, y cada instancia de la aplicación es un objeto de esta clase. Las clases en MATLAB tiene tres tipos de componentes principales:

- **Bloques de propiedades.** Definen las propiedades que almacenan la información para cada objeto de la clase.
- **Bloques de métodos.** Contienen las funciones que definen las operaciones que pueden ejecutarse en cada objeto de la clase.

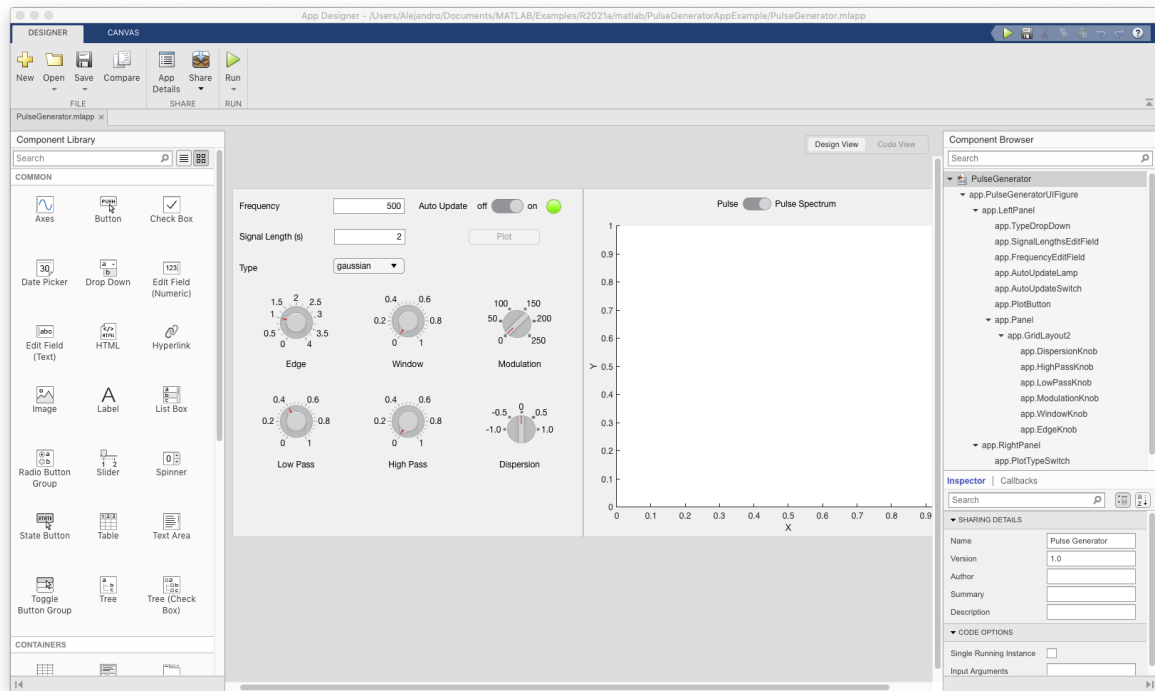


Fig. 2.29: Vista Design View de App Designer con una aplicación de ejemplo [elaboración propia]

- **Bloques de eventos.** Definen los mensajes que un objeto enviará a otras partes de una aplicación cuando algo cambie en ese objeto.

Al igual que en otros lenguajes de programación orientada a objetos (OOP), se pueden especificar atributos a las propiedades y métodos de las clases, tales como el acceso (público, privado o protegido) o la abstracción.

Las aplicaciones diseñadas mediante este entorno pueden compartirse como:

- **Apps de MATLAB.** Mediante un archivo de instalación, cualquier usuario puede instalar la aplicación en la galería de *apps* de su instalación de MATLAB.
- **Apps de escritorio.** Gracias al uso de distintos productos de MATLAB, pueden crearse aplicaciones independientes que se ejecuten en el ordenador sin necesidad de ninguna instalación de MATLAB.
- **Apps web.** MATLAB permite alojar aplicaciones desarrolladas con App Designer en un servidor de forma que los usuarios puedan acceder a ellas a través de un navegador, sin necesidad de *software* adicional.



# Desarrollo

## 3.1. PLANTEAMIENTO

### 3.1.1. Objetivo

El propósito de este trabajo es el de diseñar un modelo virtual de un ventilador mecánico que reproduzca fielmente el comportamiento real de uno de estos dispositivos. Este modelo, además, debe cumplir dos funciones:

- Servir como punto de partida para ingenieros y técnicos, de forma que puedan simular sus propios diseños de un ventilador mecánico partiendo de un modelo ya probado y con una infraestructura funcional.
- Permitir a los usuarios interactuar con él de forma sencilla, de modo que simule una experiencia de ventilación mecánica real y sirva como medio de aprendizaje tanto para personal clínico como para ingenieros y técnicos.

### 3.1.2. Solución adoptada

En la consecución de estos objetivos, el primer paso para idear una solución es el de seleccionar la infraestructura tecnológica más adecuada. Puesto que el propósito del proyecto es el de desarrollar un modelo virtual, sin interacción con señales físicas externas, se prescinde del uso de *hardware* específico optando por una solución basada en un ordenador personal, al igual que para el proceso de desarrollo. Dicho esto, el *software* elegido es MATLAB, por sus numerosas capacidades ya mencionadas en relación con la elaboración y simulación de sistemas.

Con respecto al desarrollo del modelo, dentro de MATLAB se utiliza la herramienta Simulink, que ofrece un entorno gráfico potente y de fácil manejo para la realización de esta actividad. Además, con el objetivo de reproducir fielmente los distintos elementos que componen un ventilador mecánico, así como el sistema respiratorio humano, se toma como base el modelo proporcionado por The MathWorks, Inc. Esta representación de un VM y un sistema respiratorio, gracias a los distintos dominios físicos que integra con las librerías de Simscape, modela de la forma más realista posible el comportamiento del flujo de aire en su interacción con los diferentes elementos del circuito y el entorno.

Por último, cumpliendo con el objetivo de hacer de esta solución una herramienta docente para los sanitarios, se utiliza el entorno App Designer del mismo MATLAB para la realización de una interfaz gráfica que permita la interacción con el modelo en Simulink de forma sencilla e intuitiva.

En cuanto al funcionamiento del ventilador, y en línea con este objetivo didáctico, se desarrollan cuatro modos de ventilación:

- **VC-CMV y PC-CMV.** Modos de acceso a la ventilación controlada por volumen y por presión, respectivamente. Permiten familiarizarse con estas dos técnicas a un nivel elemental.
- **PSV.** Modo controlado por presión con sincronización de los esfuerzos del paciente, ampliamente utilizado en casos clínicos reales. Se dispone, además, de datos experimentales para la validación del modelo.

- **PAV.** Modo más innovador, no disponible en todos los ventiladores comerciales. Acerca al usuario a un tipo de ventilación más distintivo.

## 3.2. MODELO EN SIMULINK

### 3.2.1. Funcionamiento general

El alcance del proyecto en cuanto al modelo del ventilador se ciñe mayoritariamente al diseño del algoritmo de control, el cual es el encargado de generar las señales adecuadas para hacer funcionar el VM en el modo correspondiente. Por tanto, el elemento de mayor interés en el modelo de partida es el bloque *Control Signal* (véase apartado 2.2.1.1). Los componentes físicos de la librería Simscape, en principio, quedan relegados a un segundo plano, puesto que el modelo ya ha sido diseñado con suficiente nivel de detalle y rigurosidad por ingenieros expertos en los distintos dominios físicos que se integran.

Del mismo modo, los elementos que forman parte del sistema respiratorio son modificados lo menos posible, salvo aquellos parámetros que tienen implicación directa en el buen desempeño del simulador.

Dicho esto, la implementación de los cuatro modos de ventilación considerados se realiza mediante el desarrollo de cuatro modelos de Simulink distintos. Esto tiene su fundamento principalmente en una limitación en el uso de los elementos de la librería Simscape; sin embargo, lejos de suponer un inconveniente, ha resultado comportar ciertas ventajas.

La limitación está a la hora de alternar distintos elementos de Simscape en un mismo circuito. Para el modelo VC-CMV, se utiliza como mecanismo de accionamiento el dispositivo incluido en el modelo de partida, el elemento *Volumetric Supply*, que proporciona un flujo de aire igual al valor de su señal de entrada. Por contra, para los otros modos (de control de presión), se utiliza un dispositivo similar pero que asegura que la presión en su salida sea igual al valor de su señal de entrada. La alternancia entre estos dos dispositivos en el mismo circuito ventilatorio no fue conseguida con las soluciones disponibles, por lo que se optó por un esquema de modelos referenciados.

Tal como se puede observar en la Fig. 3.1, en un primer nivel del modelo global *Simulator*, se dispone de un bloque *Variant Model* cuyo detalle se ilustra en la Fig. 3.2. En esta última se observa que existen otros cuatro elementos, uno por cada modo de ventilación a implementar. Cada bloque de estos representa un modelo de Simulink con entidad propia, y cada uno de ellos dispone de los elementos propios del ventilador, del sistema respiratorio y de su pertinente algoritmo de control. Nótese que cada uno de ellos dispone de un puerto de entrada y tres de salida, que se corresponden con las señales que llegan y salen del *Variant Model* en el nivel superior. En cada momento, solo uno de los modelos es referenciado (en este caso el denominado *VC\_CMV*, que se muestra resaltado) en base al valor de una variable del tipo *variant*; los elementos del tipo *inport* (*Pmus*(1)) y *outport* (*Flow*(1), *Volume* (2) y *Paw*(3)), que en este contexto aparecen sin uniones físicas visibles, establecen las conexiones con el modelo referenciado en cada momento.

De vuelta al nivel superior, se visualizan 4 señales:

- ***Pmus***. Señal generada algorítmicamente para simular el esfuerzo del paciente.
- ***Flow***. Señal obtenida de los sensores que representa el flujo de aire medido en boca.
- ***Volume***. Señal obtenida de los sensores que representa el volumen de aire medido en boca.
- ***Paw***. Señal obtenida de los sensores que representa la presión medida en boca.

El elemento *Graphics* es un *scope*, que permite visualizar las señales que le llegan, mientras que el bloque *Pmus* es el subsistema que genera la señal de esfuerzo del paciente, del cual se hablará más tarde. Los otros cuatro elementos, que indican la ruta a un archivo *.mat*, tienen su fundamento en el diseño de la GUI, la cual también se comentará en un apartado posterior.



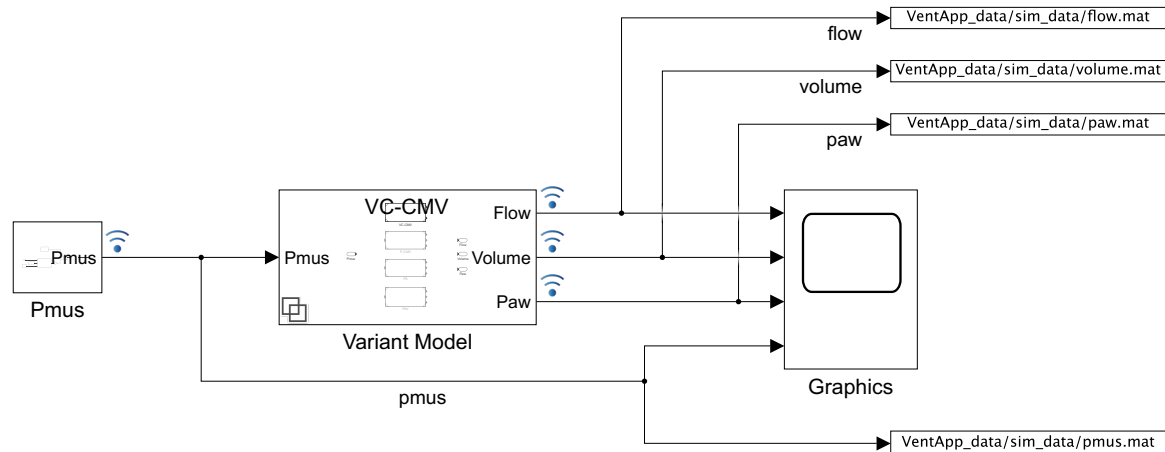


Fig. 3.1: Nivel superior del modelo *Simulator* [elaboración propia]

La variable de tipo *variant* es un objeto del entorno Simulink que contiene una condición evaluable a verdadero o falso. En caso de que la condición sea verdadera, el modelo referenciado asociado a ella se selecciona dentro del entorno variante. Cada modelo debe estar asociado a un objeto de este tipo distinto; en caso de que distintos objetos evalúen la misma condición y, por tanto, se produzca la selección de dos modelos en un mismo bloque variante, el sistema lanzará un error. La selección de los modelos para este caso se realiza mediante la ejecución de las instrucciones del Listado 3.1 en el editor de MATLAB, que inicializa las variables indicadas en el espacio de trabajo base, al cual se tiene acceso desde Simulink.

Listado 3.1: Definición de variables para la selección del modo de ventilación

```

%% Select modes
mode = 1; % VC-CMV(1), P-CMV(2), PS(3), PAV(4)
vVCCMV = Simulink.Variant('mode == 1');
vPC = Simulink.Variant('mode == 2');
vPS = Simulink.Variant('mode == 3');
vPAV = Simulink.Variant('mode == 4');

```

Con esta configuración, se puede alternar entre los distintos modos de ventilación desde la ejecución de un único modelo global (*Simulator*), con tan solo cambiar el valor de la variable *mode*. Además, se consigue un compromiso entre independencia de los modelos y modularidad. Independencia porque los cambios en un modelo no afectan a los demás, por lo que la experimentación en un modo se puede dar sin comprometer el comportamiento de los otros; y modularidad porque tanto la señal de entrada como las de salida son comunes a todos los modelos. Además, los parámetros que afectan a todos los modos, como pueden ser la resistencia y elastancia del paciente, están configurados en los modelos de forma que su valor dependa de las mismas variables globales definidas en el espacio de trabajo base, por lo que modificar el valor de estas supone modificar el parámetro asociado en todos los modelos. Por último, la inclusión de nuevos modos de ventilación puede darse de forma muy sencilla bajo estos mismos principios de independencia y modularidad.

A propósito de seguir con una estructura modular, el nivel más alto de cada uno de los cuatro modelos se ha organizado tal como se muestra en la Fig. 3.3. En ella se observan 3 subsistemas:

- **Control.** Incluye los elementos necesarios para generar las señales de control de cada modo.
- **Ventilator.** Agrupa los componentes propios del sistema del ventilador. Incluye los sensores que generan las señales accesibles mediante las etiquetas  $\dot{V}_{lung}$ ,  $V_{tidal}$  y  $Paw$ , que serán las salidas del bloque *Variant Model* del modelo *Simulator*.

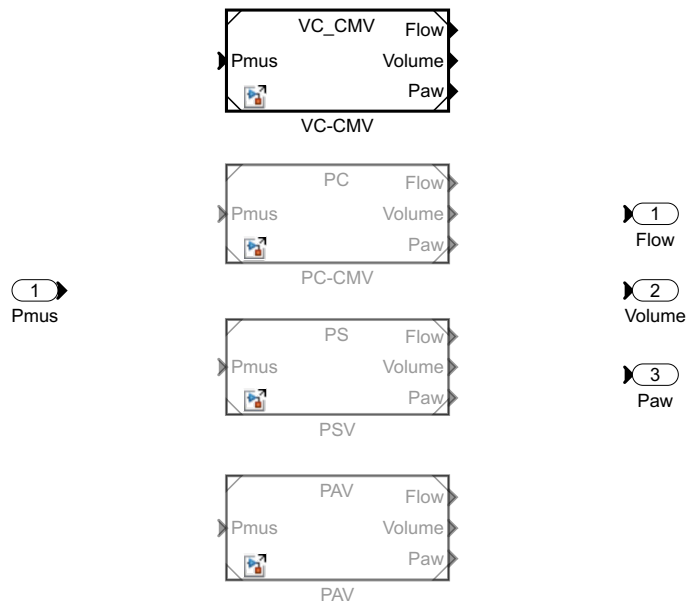


Fig. 3.2: Detalle del bloque *Variant Model* en el modelo *Simulator* [elaboración propia]

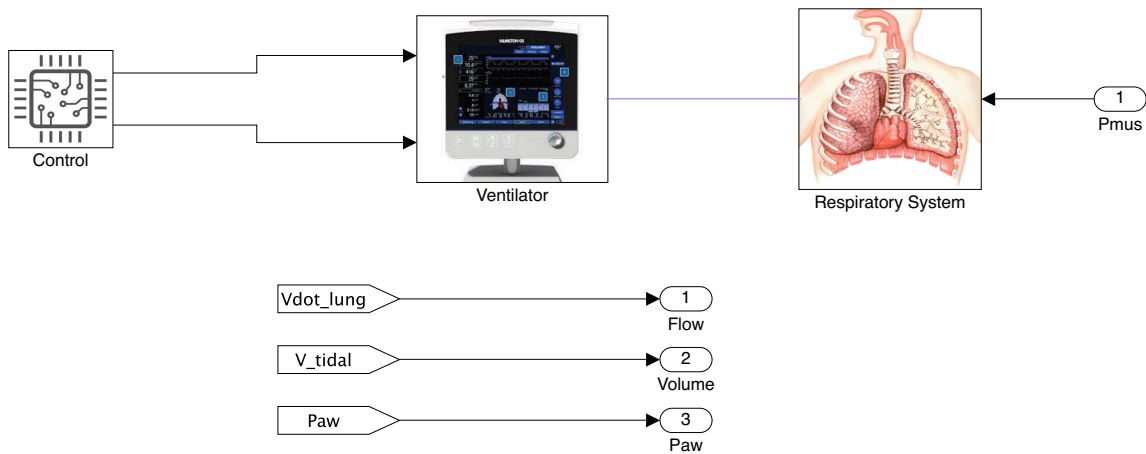


Fig. 3.3: Nivel superior del modelo que implementa el modo VC-CMV [elaboración propia]

- **Respiratory System.** Agrupa los componentes propios del sistema respiratorio. La señal *Pmus* generada en el nivel superior del *Simulator* es una entrada de este subsistema.

Las tres etiquetas asociadas a los tres elementos *outport* envían las señales generadas en los sensores (flujo, volumen y presión en boca) al nivel superior, en este caso el bloque *Variant Model*, que a su vez dispone de sus elementos *outport* virtuales para acceder a las señales desde el primer nivel del modelo *Simulator*.

### 3.2.1.1. Sistema Respiratorio

Dado que el subsistema que modela el sistema respiratorio del paciente es idéntico para todos los modelos, se detalla el funcionamiento de este previo a la descripción de cada uno de los algoritmos implementados.

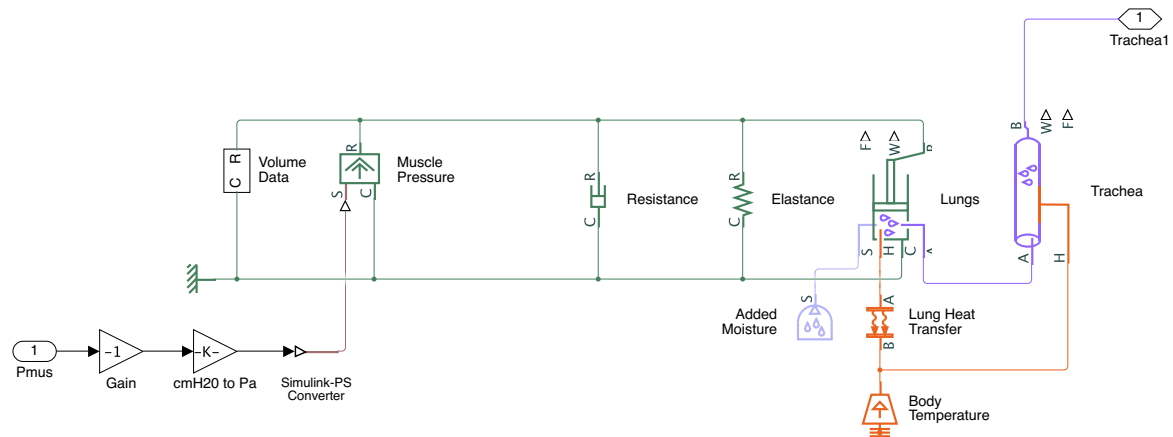


Fig. 3.4: Detalle del subsistema *Respiratory System* [elaboración propia]

Tal como se puede observar en la Fig. 3.4, este subsistema incluye los componentes del modelo de partida que simulan tanto la tráquea como el conjunto de los pulmones, con todos los elementos mecánicos, térmicos y de aire húmedo correspondientes. Sin embargo, se ha sustituido el elemento *PS Constant* por una serie de elementos que se describen a continuación:

- **Pmus.** Elemento del tipo *inport* que actúa de interfaz entre los distintos niveles de un modelo. En este caso, la señal de salida de este componente equivale a la señal de entrada al subsistema *Respiratory System*, que a su vez proviene de otro elemento *inport* con el mismo nombre.

Esta señal se remonta al subsistema *Pmus* del primer nivel del modelo *Simulator*, que tal como se había introducido anteriormente, es el encargado de generar esta señal de esfuerzo del paciente. El detalle de este subsistema se ilustra en la Fig. 3.5. En ella podemos observar dos ramas

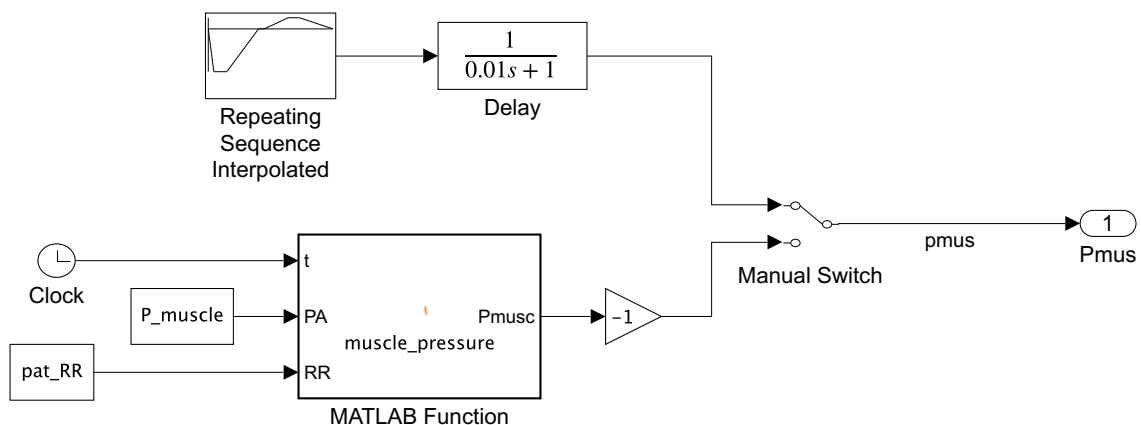


Fig. 3.5: Detalle del subsistema *Pmus* en el modelo *Simulator* [elaboración propia]

conectadas por un elemento *Manual Switch*. Este interruptor permite conmutar entre los dos elementos generadores de la señal de esfuerzo inspiratorio (véase Fig. 3.6):

- **Repeating Sequence Interpolated.** Este elemento genera una señal que se repite en el tiempo a partir de un vector de salidas y otro vector de tiempos. Además, también requiere de un tiempo de muestreo, que se define mediante la variable  $h$ , utilizada para todos los sistemas discretos del modelo. La definición de estos vectores se muestra en el Listado 3.2.

Listado 3.2: Vectores de valores de salida y de valores de tiempo del bloque *Repeating Sequence Interpolated*

```
[0 -insp_P -insp_P 0 0 exp_P exp_P 0].'
```

```
[0, insp_increase*60/pat_RR, (insp_increase+insp_hold)*60/pat_RR, (
    insp_increase+insp_hold+insp_release)*60/pat_RR, (insp_increase+insp_hold+
    insp_release+insp_pause)*60/pat_RR, (insp_increase+insp_hold+insp_release+
    insp_pause+exp_increase)*60/pat_RR, (insp_increase+insp_hold+insp_release+
    insp_pause+exp_increase+exp_hold)*60/pat_RR, (insp_increase+insp_hold+
    insp_release+insp_pause+exp_increase+exp_hold+exp_release)*60/pat_RR].'
```

En el vector de valores de salida se utilizan dos variables, correspondientes a las presiones durante la inspiración ( $insp\_P$ ) y la espiración ( $exp\_P$ ), con signos contrarios y en cmH<sub>2</sub>O. El vector de tiempos utiliza ocho variables: tres para definir el tiempo inspiratorio, tres para el tiempo espiratorio, una para la pausa entre ambas fases y otra para definir el tiempo de cada respiración. Las tres variables de cada fase representan el tiempo de subida (ataque o *increase*) hasta la presión correspondiente, el tiempo de mantenimiento a este nivel (*hold*) y el tiempo de bajada hasta la presión cero (*release*). Estas siete variables se definen como un porcentaje del tiempo de una respiración completa, por lo que la suma debe dar siempre uno. La variable  $pat\_RR$  se asocia con el valor de frecuencia respiratoria (*respiratory rate*) del paciente, en respiraciones por minuto.

El elemento *Delay* multiplica la señal por la función de transferencia mostrada, lo cual suaviza ligeramente la curva generada. Esta señal es la utilizada en el funcionamiento normal del simulador.

- **MATLAB Function.** Este bloque implementa una función escrita en el lenguaje de programación de MATLAB, en este caso la denominada *muscle\_pressure*. Esta función genera una señal de esfuerzo inspiratorio, en términos también de presión, que busca aproximarse lo máximo posible a la forma de onda de las señales generadas por los pacientes en un ensayo clínico real. Los datos obtenidos de este experimento son los que se usan en la validación de este trabajo, por lo que la  $P_{mus}$  utilizada en este proceso es la generada mediante esta rama.

Esta función, que ha sido facilitada por la tutora del presente proyecto, toma como parámetros los valores de presión inspiratoria ( $P_{muscle}$  en este caso), y frecuencia respiratoria del paciente, y el tiempo instantáneo de la simulación. El bloque de ganancia que multiplica la señal por  $-1$  se justifica con el objetivo de mantener la concordancia con la primera rama y a su vez por cuestiones de representación, dado que el esfuerzo inspiratorio genera una presión negativa en la vía aérea.

- **Ganancias.** Ambos bloques multiplicadores tienen el objetivo de adecuar la señal  $P_{mus}$  al modo de funcionamiento del elemento *Muscle Pressure*. La primera de ellas vuelve a invertir la señal de forma que produzca un desplazamiento positivo en la red mecánica y la segunda cambia las unidades de medida de centímetros de agua a pascales.
- **Simulink PS Converter.** Convierte la señal de Simulink en una señal física, adecuada para su uso con el bloque *Muscle Pressure*.
- **Trachea1.** Este elemento es análogo al *inport*, solo que conecta señales físicas, en este caso el aire húmedo que sale de la máscara.

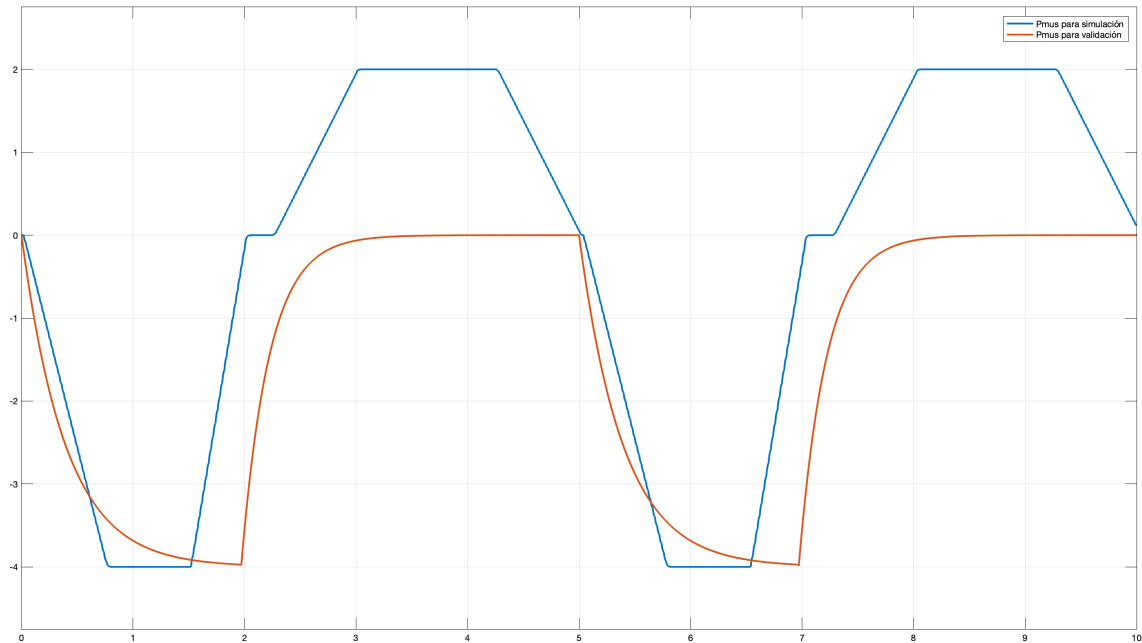


Fig. 3.6: Distintos tipos de señales de esfuerzo inspiratorio generadas por el subsistema *Pmus* ( $insp\_P = 4$ ;  $insp\_increase = 0,15$ ;  $insp\_hold = 0,15$ ;  $insp\_release = 0,10$ ;  $insp\_pause = 0,05$ ;  $exp\_P = 2$ ;  $exp\_increase = 0,15$ ;  $exp\_hold = 0,25$ ;  $exp\_release = 0,15$ ;  $pat\_RR = 12$ ;  $P\_muscle = 4$ ) [elaboración propia]

Destacar también de este modelo que el valor de los elementos *Resistance* y *Elastance* se define mediante las variables  $R\_respiratory$  y  $E\_respiratory$ , en  $cmH_2O/(L/s)$  y  $cmH_2O/L$ , respectivamente. De esta forma, cambiar estos valores en el espacio de trabajo base modifica la impedancia del paciente para todos los modelos.

### 3.2.1.2. Ventilador

Asimismo, el subsistema del ventilador, es prácticamente idéntico para los cuatro modelos realizados, por lo que en este apartado se detallan las generalidades y modificaciones sobre el modelo de partida que adoptan los cuatro modos de ventilación.

Tal como se observa en la Fig. 3.7, en el subsistema *Ventilador* se han agrupado los elementos que representan desde la entrada de aire al circuito hasta su salida, incluyendo la máscara y los sensores ubicados en ella. Las señales de control llegan mediante dos elementos *inport* como señales de Simulink; la señal que controla la apertura de la válvula espiratoria está definida con valores entre 0 y 1, siendo el valor de la ganancia *Max Valve Area* de  $\pi \cdot D\_tube^2/4$ , es decir, el máximo área de la válvula. Para la conexión de la salida del conjunto máscara/sensores con la entrada del sistema respiratorio se utiliza el elemento *Trachea*, que como es de esperar, es un elemento del tipo *outport* para señales físicas.

### Válvula PEEP

Nótese que la *Check Valve 2* ha sido sustituida por una *PEEP Valve*. Este elemento pretende simular una válvula PEEP mecánica, como la mostrada en la Fig. 3.8. Este tipo de válvulas regulan la presión mediante un resorte cuya tensión puede ser ajustada manualmente, relacionándose con un nivel de presión que se debe vencer para permitir el paso del aire. En el ejemplo mostrado, el ajuste es de 5  $cmH_2O$ , pero se indica que la presión en el tubo espiratorio (antes de la válvula) debe ser mayor que

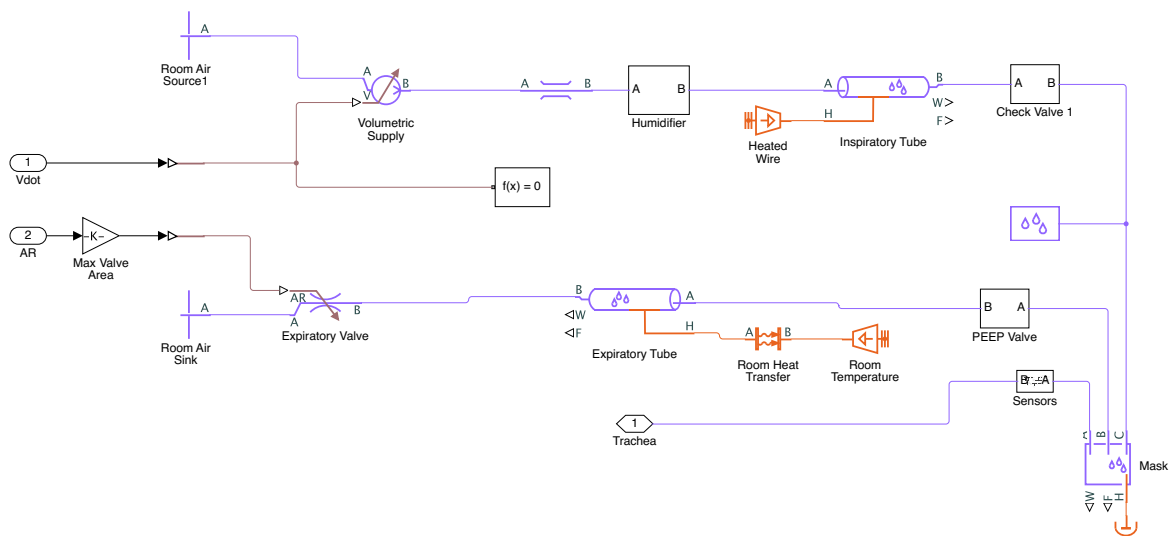


Fig. 3.7: Detalle del subsistema *Ventilator* en el modelo que implementa el modo VC-CMV [elaboración propia]

10 cmH<sub>2</sub>O para permitir el flujo; esto es debido a los otros 5 cmH<sub>2</sub>O, que se corresponden con la presión en la rama espiratoria después de la válvula, la cual es regulada mediante la apertura de la válvula espiratoria [25].

La solución adoptada para alcanzar el nivel de PEEP deseado incluye el control de la apertura de la válvula espiratoria mediante una estrategia todo-nada. De esta forma, durante la inspiración dicha válvula se encuentra cerrada, y por lo tanto no es posible el flujo por la rama espiratoria independientemente del ajuste de la válvula PEEP. Durante la exhalación, la válvula espiratoria se abre por completo, por lo que la presión a vencer después de la *PEEP Valve* será únicamente la causada por las resistencias del circuito aguas abajo, que pueden considerarse despreciables. Por tanto, el ajuste de esta válvula se corresponderá con el valor de PEEP deseado. Además, destacar que este elemento actúa también como válvula antirretorno, por lo que la funcionalidad de la *Check Valve 2* no ha sido eliminada.

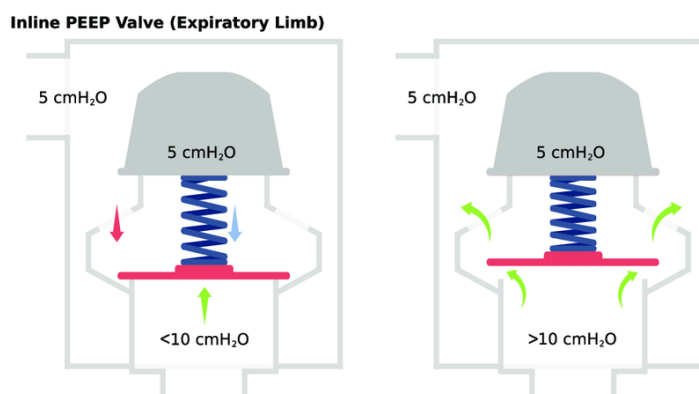


Fig. 3.8: Esquema de funcionamiento de una válvula PEEP en la rama espiratoria [25]

Para describir la forma de implementar este funcionamiento, es necesario referirse al detalle del elemento *Check Valve* (Fig. 2.26). Este subsistema tiene entre sus componentes el elemento *Pressure to Area*, del tipo *PS Lookup Table*. Esta clase de bloque actúa como un convertidor para una señal física de

entrada, donde la relación entre esta y la salida viene determinada por una tabla de consulta o *lookup table*. Para el caso de una de las *Check Valve*, este bloque relaciona la presión medida en el punto del circuito (en pascales) con un área de paso de la válvula (en m<sup>2</sup>): a mayor presión, mayor grado de apertura. La tabla de consulta se define en los parámetros del bloque (Fig. 3.9) mediante una serie de puntos determinados por los vectores *Table grid vector* y *Table values*. En estos vectores intervienen cuatro variables: diferenciales de presión para conseguir la mínima y máxima aperturas de la válvula ( $p_{crack}$  y  $p_{max}$ , respectivamente), y las áreas mínima y máxima de la válvula ( $A_{leak}$  y  $A_{max}$ , respectivamente). Para el caso de la *PEEP Valve*, estos vectores se han especificado de la siguiente manera (Listado 3.3), en las mismas unidades:

Listado 3.3: Valores de presión y área de la tabla de consulta del bloque *Pressure to Area* en la *PEEP Valve*

```
[-0.2 -0.1 0 0.2 0.5 0.8 1 1.1 1.2]*(p_max - max(p_crack, PEEP*cmH2O_to_Pa)) + max(p_crack, PEEP*cmH2O_to_Pa)
```

```
[0 0 0 0.2 0.5 0.8 1 1 1]*(A_max - A_leak) + A_leak
```

Las variables *PEEP* y *cmH2O\_to\_Pa* definen, respectivamente, el valor de PEEP en cmH2O establecido y el factor de conversión de estas unidades a pascales (98,0665). Las curvas que definen la relación entre la presión y la apertura de las válvulas *Check Valve* y *PEEP Valve* se muestran en la Fig. 3.10.

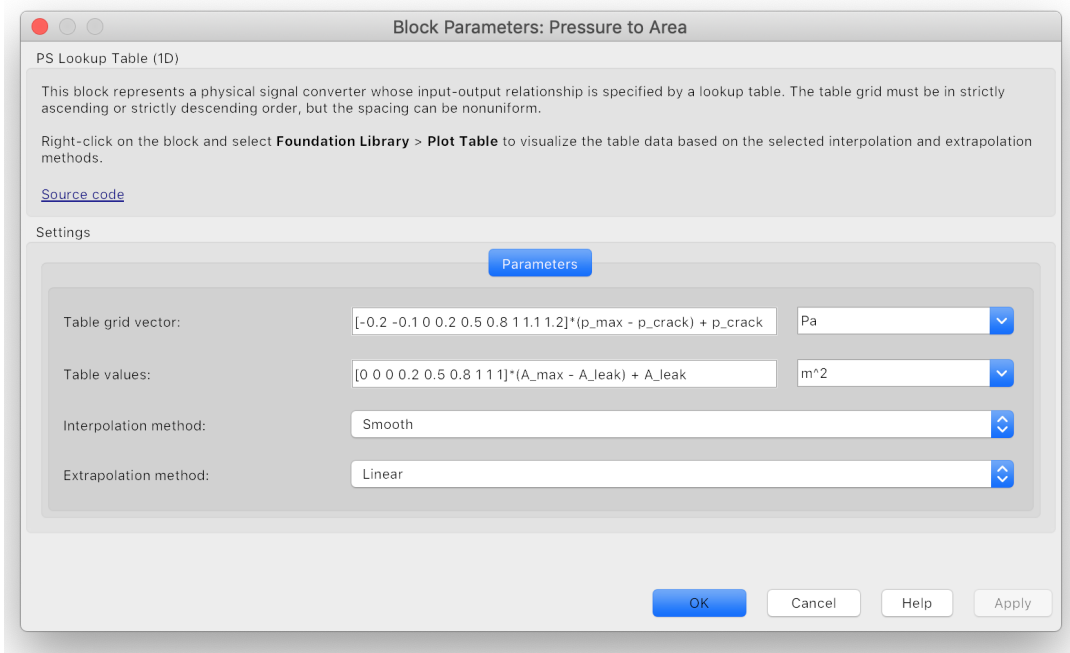


Fig. 3.9: Configuración del bloque *Pressure to Area* en una *Check Valve* [elaboración propia]

Con este método se consigue que, una vez la presión en la vía aérea descienda hasta el nivel de PEEP establecido durante la espiración, la *PEEP Valve* se cierre y se mantenga por tanto este nivel de presión en boca.

Una alternativa a esta solución es el control de la PEEP directamente mediante la regulación de la apertura de la válvula espiratoria, lo cual comporta algunas desventajas. Primeramente, el cierre de la válvula no puede realizarse de forma que, cuando el nivel de presión en la vía aérea alcance el de la PEEP, esta se cierre por completo. Esto ha sido probado que causa un *efecto rebote* en la presión en boca, que se incrementa considerablemente por encima del nivel de PEEP. La solución pasaría por cerrar el lazo y ajustar el grado de apertura de la *Expiratory Valve* en función de la presión en boca medida; lo cual supone diseñar un controlador adecuado y, además, consumir más recursos

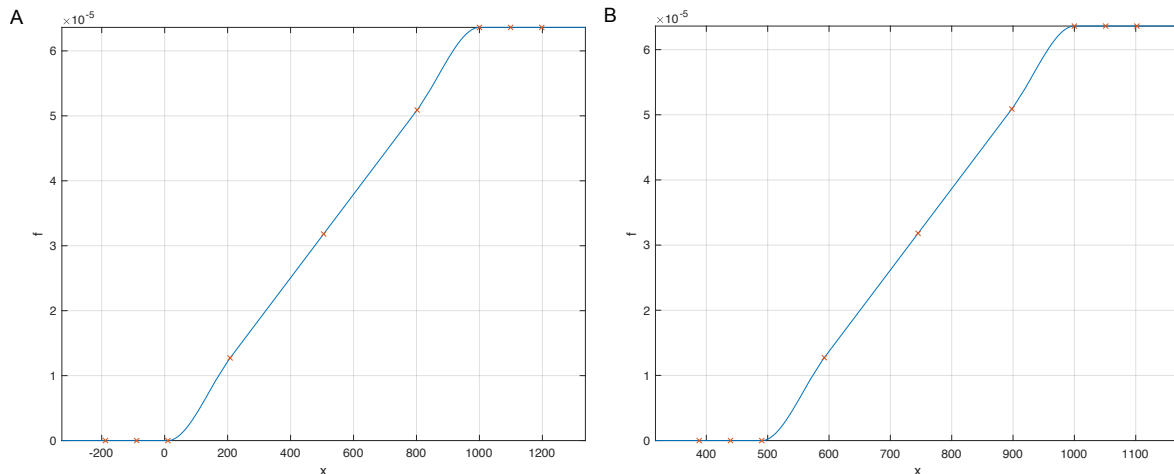


Fig. 3.10: Relación entre la presión diferencial y el área de apertura de las *Check Valve* (A) y *PEEP Valve* (B) (PEEP = 5 cmH<sub>2</sub>O = 490.3325 Pa) [elaboración propia]

computacionales. Como ventaja se encuentra que el ajuste de la PEEP, en un caso de ventilador real, podría hacerse automáticamente desde el panel de programación del VM, en lugar de manualmente como sería implementando la solución finalmente adoptada. Dado que el resultado tangible de este proyecto es un simulador, se sacrifica la alternativa cuya única ventaja no es aprovechable en este contexto, adoptando la solución más sencilla, precisa y económica computacionalmente, y que, además, aporta un conocimiento extra en cuanto a la mecánica de los ventiladores.

## Sensores

Sobre el subsistema *Sensors* del modelo de partida también se han realizado algunas modificaciones. En comparación con la Fig. 2.27, la Fig. 3.11 ilustra dos nuevas señales:

- **$V\_tidal$ .** Esta señal representa la variación de volumen a lo largo de todo el proceso ventilatorio. Se obtiene mediante el uso de un bloque *Integrator*, el cual integra la señal de flujo  $Vdot\_lung$ . Además, el bloque integrador es reseteado al final de cada respiración, siguiendo el ejemplo de los ventiladores reales, de forma que no se acumule error en la integración de la señal de flujo y se vea claramente el volumen tidal entregado en cada respiración. El reseteo se da cuando se detecta un flanco positivo en la salida de la puerta *AND*, la cual produce un 1 cuando el flujo es mayor que 0 (fase inspiratoria) y cuando este se encuentra en ascenso durante un tiempo prudente (para evitar rebotes); en el caso del modo VC-CMV, se comprueba el ascenso de la señal de presión en lugar de la de flujo. Esta señal será una de las salidas globales del modelo, junto con el flujo y la presión en la vía aérea (que se ha renombrado como  $Paw$ , del inglés *airway pressure*). La señal es similar a aquella generada por el subsistema *Volume Data*, salvo que esta última parte de la condición inicial de volumen en los pulmones o capacidad funcional residual (FRC) y no se resetea en ningún momento. Este dato, aunque es accesible en el modelo mediante la variable  $FRC$ , no se manipula durante las simulaciones dado que no tiene influencia en los resultados. Con todo, la señal  $V\_lung$  generada por el *Volume Data* no se utilizará en el modelo, puesto que simula una experiencia de medición poco realista. La manera habitual de monitorizar la variación de volumen es mediante la integración de la señal de flujo.
- **$V\_vent$ .** Esta señal representa el volumen ventilado durante la fase inspiratoria. Se obtiene integrando la señal de flujo mientras esta es positiva, es decir, durante la inspiración; cuando el flujo es igual o menor que cero, la espiración comienza y se resetea el integrador, haciendo que



la  $V_{vent}$  valga cero durante esta fase. El motivo de esta señal es su utilidad en los algoritmos de control, concretamente, para los modos VC-CMV y PAV.

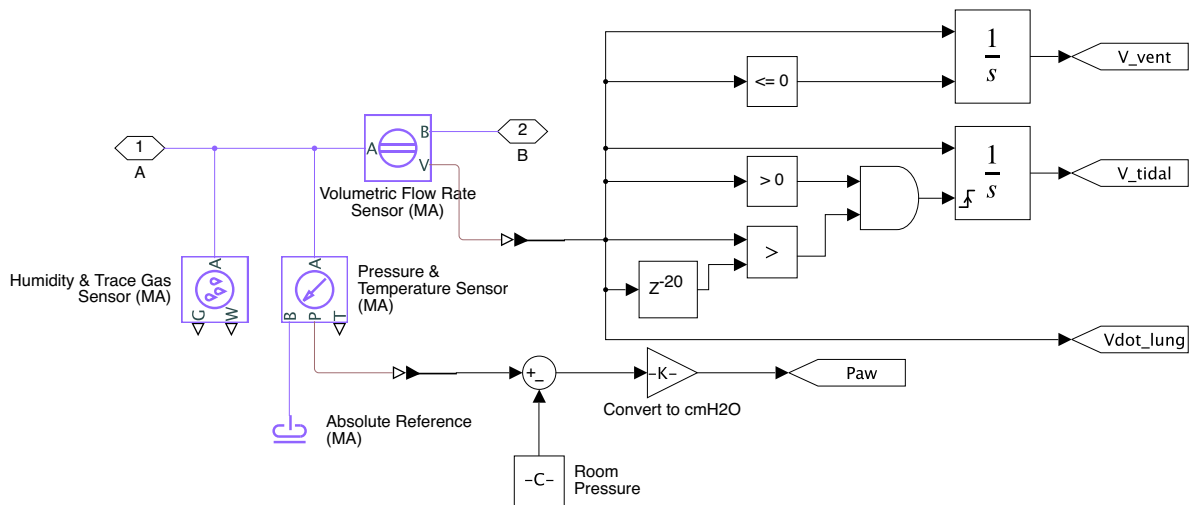


Fig. 3.11: Detalle del subsistema *Sensors* en el modelo que implementa el modo VC-CMV [elaboración propia]

### 3.2.2. VC-CMV

En cuanto al algoritmo de control del modo VC-CMV, este se implementa mediante los bloques mostrados en la Fig. 3.12. El funcionamiento se describe a continuación.

Los dos bloques *outport Vdot\_control* y *Area\_Valve* son los encargados de conectar las señales generadas en este subsistema con los elementos *Volumetric Supply* y *Expiratory Valve* del *Ventilator*, a través de

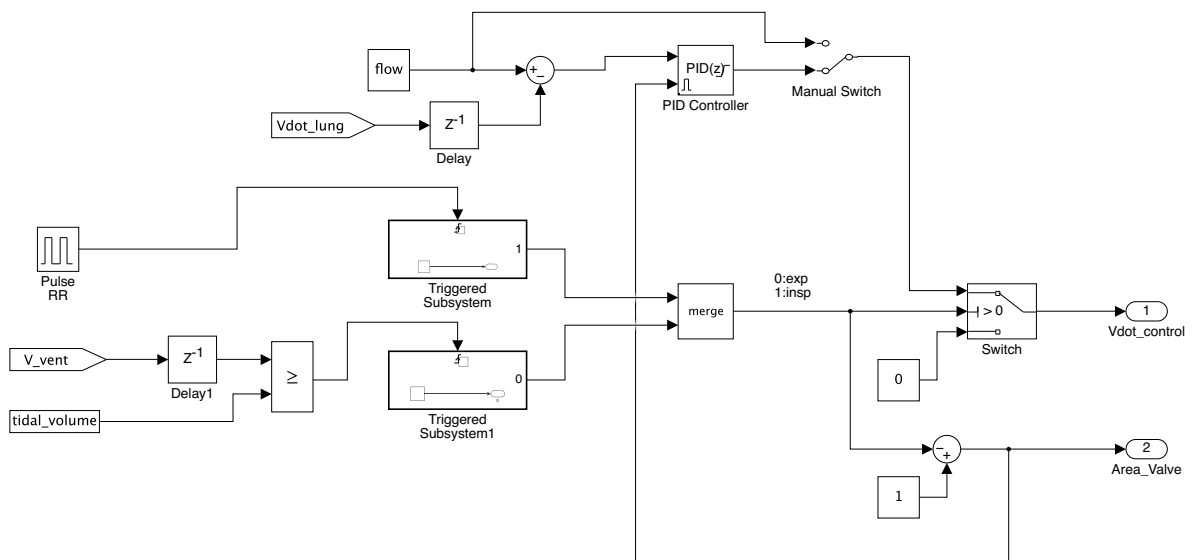


Fig. 3.12: Detalle del subsistema *Control* en el modelo que implementa el modo VC-CMV [elaboración propia]

sus correspondientes bloques *inport*. Estas señales generadas son la consigna de flujo a conseguir y la apertura de la válvula espiratoria.

Para la señal de flujo, de tipo escalón, se utiliza un elemento *Switch* para alternar entre un valor de flujo positivo y un valor de 0 L/s. Este interruptor cambia el origen de la señal en base a una condición booleana: si el valor de la señal conectada a su segunda entrada satisface el criterio de ser mayor que 0, la salida del *Switch* se conecta a la primera entrada (rama superior); en caso contrario, el valor de la señal de flujo será igual a cero, pues la tercera entrada del interruptor está conectada a un bloque constante con valor 0.

La rama superior está compuesta por varios elementos que conforman un esquema de control en lazo cerrado. Por un lado, se tiene el bloque constante con el valor de *flow*. Esta variable, calculada en base al volumen tidal y el tiempo inspiratorio determinados (ver ecuación 2.6), es el valor de flujo que se pretende proporcionar al paciente de manera continua durante la inspiración. Para asegurar que el flujo recibido por el paciente es igual al deseado, se cierra el lazo de control. Esto se realiza gracias al elemento *From* con la etiqueta *Vdot\_lung*, que se recuerda se asocia con el flujo medido por los sensores en boca. A esta señal es necesario aplicarle un retraso o *Delay* con valor de al menos un intervalo temporal; esto es debido a que, en caso contrario, se produciría un bucle algebraico y consecuentemente un error, al depender el flujo medido en cierto momento de sí mismo, en el mismo instante. Con estas dos señales, de consigna deseada y flujo real conseguido, se restan y se consigue la señal de error, que sirve como entrada del *PID Controller*.

Este elemento implementa un controlador del tipo PID. Este tipo de controladores proporcionan una salida en función de la señal de error y de, principalmente, tres parámetros o ganancias, una para cada término: proporcional (P), integral (I) y derivativo (D). Así pues, la señal de salida se obtiene de la siguiente manera para un PID en tiempo continuo (3.1):

$$u = P e(t) + I \int_0^t e(t)dt + D \frac{de(t)}{dt} \quad (3.1)$$

En este caso, el PID implementado es en tiempo discreto, por semejanza con los controladores actuales basados en electrónica digital y para reducir el coste computacional. El ajuste de las ganancias o *tunning* se realiza de manera experimental, en base a distintos parámetros ventilatorios. El *Manual Switch* presente a la salida del PID permite conmutar, manualmente, entre la señal generada por el controlador y la obtenida directamente de la variable *flow*, para labores de testeo del PID. En la Fig. 3.13 se puede observar la comparación entre el flujo realmente entregado con el esquema de control en lazo abierto y con el de lazo cerrado.

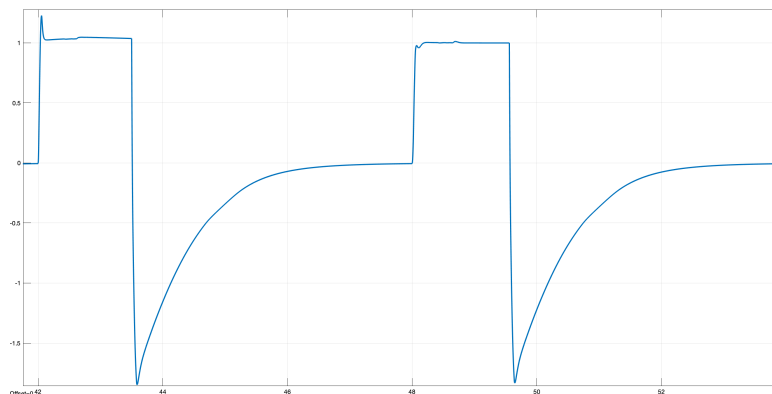


Fig. 3.13: Flujos medidos en la vía aérea con un control en lazo abierto (*izquierda*) y otro en lazo cerrado (*derecha*) en el modo VC-CMV ( $flow = 1$  L/s;  $P = 0,3$ ;  $I = 15$ ;  $D = 0,1$ ) [elaboración propia]

La señal que se evalúa para seleccionar la conexión en el *Switch* se genera de forma que valga 1 durante la fase inspiratoria y 0 durante la espiratoria. Esto se consigue mediante el uso de varios elementos, entre los que destacan el elemento *merge*. Este bloque combina dos señales en una, estableciendo el valor a su salida de la entrada más recientemente actualizada. Su uso es habitual con subsistemas ejecutados condicionalmente, como lo son ambos *Triggered Subsystem*. Estos dos subsistemas se ejecutan cuando la señal de entrada correspondiente sufre un flanco positivo; en su interior solamente disponen de un bloque constante conectado a un elemento *outport* (véase Fig. 3.14), lo cual provoca que, cuando uno de ellos es ejecutado, la señal a la salida del bloque *merge* tome el valor del bloque constante de dicho *Triggered Subsystem* (0 ó 1).



Fig. 3.14: Detalle del elemento Triggered Subsystem del subsistema Control en el modelo que implementa el modo VC-CMV [elaboración propia]

El elemento *Pulse RR* es un generador de pulsos configurado para ejecutarse con un periodo igual a la frecuencia respiratoria ajustada en el ventilador. Esto provoca que en los flancos positivos, que marcan el inicio de la inspiración, la salida del bloque *merge* tome el valor 1. Por otra parte, los elementos que condicionan la ejecución del *Triggered Subsystem1* comparan continuamente la señal medida de volumen ventilado con el volumen tidal establecido para la ventilación. Cuando se alcanza un volumen ventilado igual al prefijado para cada respiración, a la salida del elemento *merge* se escribe un 0, indicando el inicio de la espiración y haciendo entregar un flujo de 0 L/s.

La señal que controla la apertura de la válvula espiratoria, que como se ha comentado previamente se hace mediante una estrategia todo-nada, simplemente es la señal a la salida del *merge* invertida. Durante la inspiración, la válvula se encuentra cerrada (0), mientras que durante la fase espiratoria, esta se abre por completo (1).

En la Fig. 3.15 se muestra el resultado de una simulación del modo VC-CMV. Se observa como el esfuerzo del paciente, en esta estrategia ventilatoria, tan solo supone una perturbación en el sistema, pues las respiraciones son todas mandatorias. También se percibe como el objetivo de volumen tidal es alcanzado, así como el de nivel de PEEP.

### 3.2.3. PC-CMV

El modelo del ventilador para los modos controlados por presión es ligeramente distinto al del modo VC-CMV (Fig. 3.16). La principal diferencia es el uso del elemento *Controlled Pressure Source (MA)*, el cual representa una fuente de energía mecánica idea en una red de aire húmedo, capaz de mantener un diferencial de presión igual al especificado mediante la señal conectada a su puerto *P*, independientemente del flujo. La señal de control se especifica en pascales, de ahí el elemento *toPa* que modela una ganancia con el valor de la conversión entre cmH<sub>2</sub>O y pascales (*cmH2O\_to\_Pa*).

La estructura del algoritmo de control del modo PC-CMV es también similar a la del modo VC-CMV (Fig. 3.17), aunque más sencilla.

Se dispone igualmente de un bloque *Switch* que alterna entre la consigna generada por el PID y el valor de presión durante la espiración, el nivel de PEEP; aunque en realidad la presión en el circuito durante la espiración es controlada por la *PEEP Valve*, se establece este nivel de presión para el elemento *Controlled Pressure Source (MA)* de forma que el circuito se mantenga presurizado en caso de perturbaciones. La señal que controla el interruptor, y por tanto indica la inspiración o espiración, se genera mediante

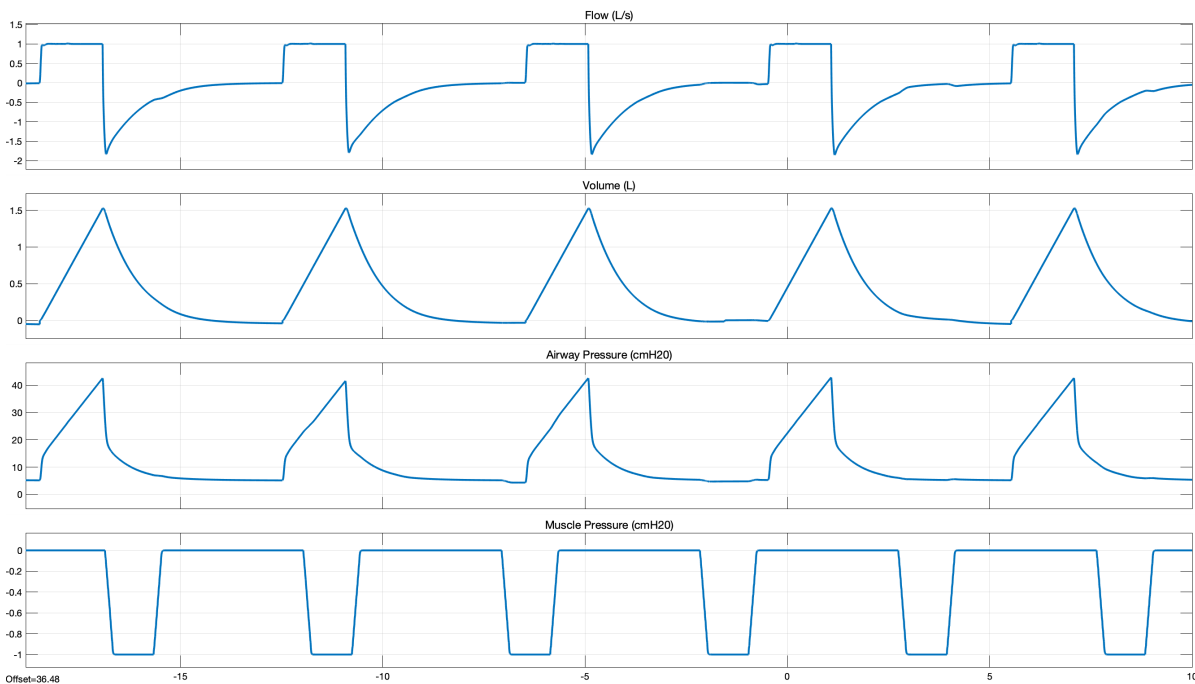


Fig. 3.15: Resultados de simulación del modo VC-CMV (FR = 10 resp/minuto;  $T_i:T_e = 0,25$ ;  $V_T = 1,5$  L; PEEP = 5 cmH<sub>2</sub>O;  $R = 9$  cmH<sub>2</sub>O/(L/s);  $E = 20$  cmH<sub>2</sub>O/L) [elaboración propia]

el bloque *Pulse Generator*, el cual produce pulsos de amplitud 1, frecuencia igual a la respiratoria y porcentaje del ancho de pulso igual a la relación  $T_i:T_e$ ; es decir, genera un 1 durante la inspiración y un 0 durante la espiración. Al igual que en el modo VC-CMV, la señal que controla la apertura de la válvula espiratoria es la opuesta a esta.

El bloque *Pressure Pattern* es, al igual que el generador de la *Pmus*, un bloque del tipo *Repeating Sequence Interpolated*. En este caso, los vectores de salida y de tiempos se establecen tal como se muestra en el Listado 3.4.

Listado 3.4: Vectores de valores de salida y de valores de tiempo del bloque *Pressure Pattern* en el modelo que implementa el modo PC-CMV

```
[PEEP Pcontrol Pcontrol PEEP PEEP]
```

```
[0, Pramp, (IE_ratio*60/respiratory_rate), (IE_ratio*60/respiratory_rate)+h, 60/  
respiratory_rate]
```

La variable *PEEP*, como es de suponer, almacena el nivel de PEEP establecido, mientras que *Pcontrol* se refiere al nivel de presión constante que se pretende alcanzar en la vía aérea del paciente. *Pramp* establece el tiempo de subida (en segundos) para la consigna de presión, desde que comienza la fase inspiratoria en el nivel de PEEP hasta que se alcanza la *Pcontrol*. La variable *IE\_ratio* indica la relación  $T_i:T_e$ , por lo que multiplicándose por el tiempo de una respiración completa ( $60/\text{respiratory\_rate}$ ) se obtiene el  $T_i$ . La variable *h* indica el tiempo de muestreo.

El lazo de control se cierra mediante la comparación de la consigna con la señal medida de presión en boca (*Paw*). El bloque *Memory* comparte el mismo propósito que los elementos *Delay* del modo VC-CMV, así como la misma funcionalidad; si bien en los bloques *Delay* puede establecerse un valor de retraso mayor que el de una muestra, en los *Memory* este parámetro no es configurable.

El controlador PID, para el caso de control de la presión, requiere ser más robusto que en el control de volumen, dado que la elastancia del sistema puede elevar considerablemente el tiempo de subida de la respuesta, provocando que el flujo máximo sea menor, al igual que el volumen tidal, debido a que no se

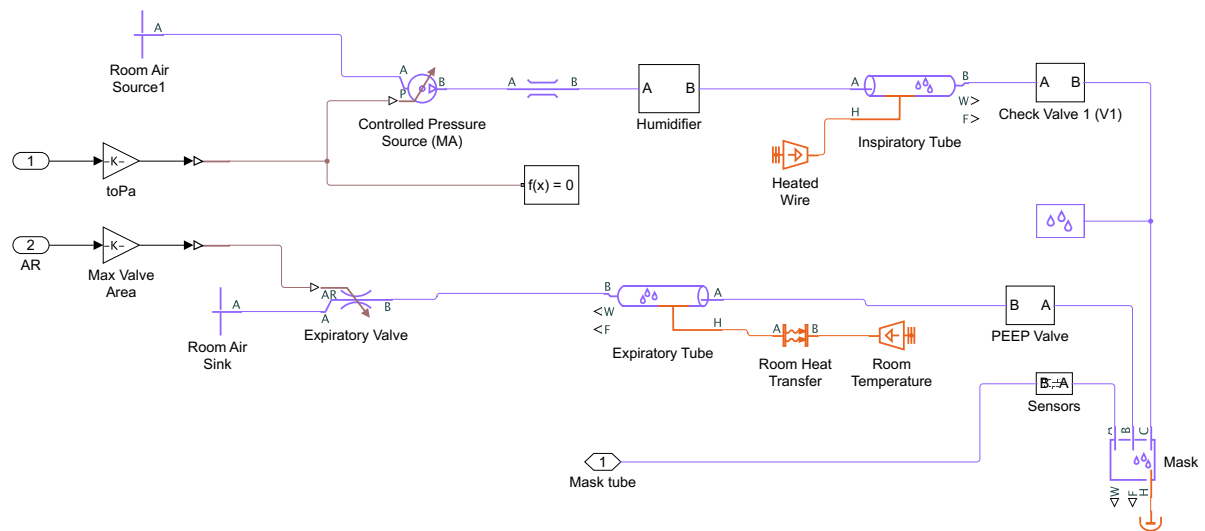


Fig. 3.16: Detalle del subsistema *Ventilator* en el modelo que implementa el modo PC-CMV [elaboración propia]

llegue a alcanzar la presión de soporte (véase 2.1.3). Es por esto que se ha aumentado considerablemente la ganancia del término integral, para alcanzar el error estacionario nulo rápidamente. En la Fig. 3.18 puede verse la comparación entre la simulación del modo PC-CMV con control en lazo abierto y en lazo cerrado. Se observa también como, al igual que para el modo VC-CMV, el esfuerzo del paciente tan solo introduce perturbaciones al sistema, las cuales son absorbidas por el controlador PID.

### 3.2.4. PSV

El modo de ventilación PSV es similar al PC-CMV, salvo que las respiraciones son espontáneas. Esto quiere decir que la consigna tiene la misma forma de presión, pero que tanto la forma de disparar como de ciclar la inspiración se vuelve más compleja. Mientras que el subsistema del ventilador no varía con respecto al PC-CMV, el nuevo esquema de control se muestra en la Fig. 3.19.

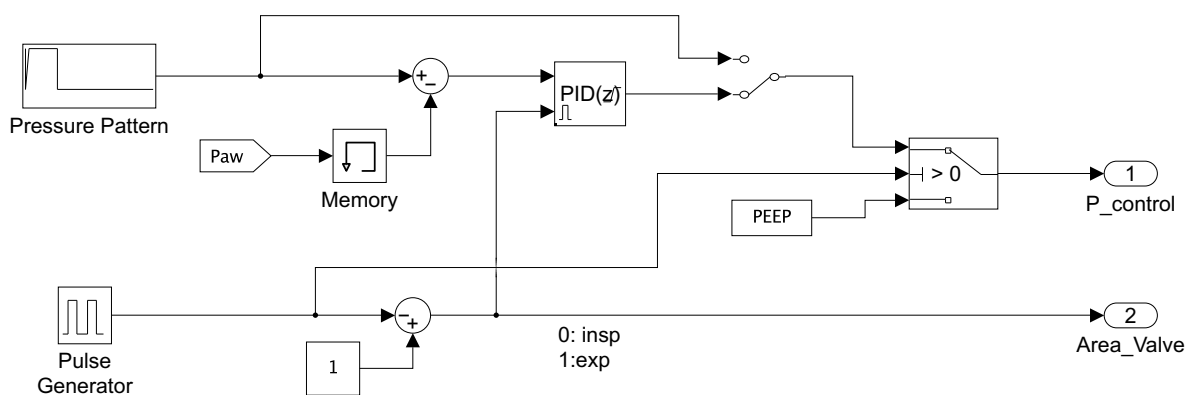


Fig. 3.17: Detalle del subsistema *Control* en el modelo que implementa el modo PC-CMV [elaboración propia]

Por un lado, se tiene el mismo funcionamiento basado en el elemento *Switch*, con una primera rama que genera la consigna en lazo cerrado y con el valor de PEEP en la tercera entrada para la fase espiratoria. La primera rama, con todo, es algo diferente. En lugar de tener un elemento *Repeating Sequence Interpolated*, se tiene un integrador. Este bloque integra una constante definida como  $(P_{control}-PEEP)/Pramp$ , lo cual describe la pendiente de la rampa de la señal de referencia. Al integrar este valor, se genera una señal creciente de acuerdo a esta pendiente; para evitar que la señal crezca indefinidamente, se limita la salida del integrador con un valor superior de  $P_{control}$ . El límite inferior se fija mediante la variable  $PEEP$ , y la salida del integrador se mantendrá a este nivel mientras la señal de reseteo del integrador se mantenga a 1 (fase espiratoria). Esto se consigue con la señal que genera el bloque *merge* y con la puerta *NOT* colocada a la entrada del puerto de *reset*; el bloque *Memory* está, nuevamente, para evitar bucles algebraicos. La realimentación del lazo se realiza de la misma forma que en el modelo anterior, y la configuración del PID se conserva con respecto a este modo, pues ambos tienen la misma dinámica.

En este modo, la inspiración puede comenzar de dos maneras, bien cuando la señal de flujo supera un umbral, o bien cuando la presión en boca disminuye por debajo de su correspondiente umbral. Para seleccionar el modo de disparo, se dispone de otro elemento *Switch* controlado por la variable *Trigger\_mode*; si el valor de esta es mayor que 0, el disparo se realiza por presión, en caso contrario, se realiza por flujo. El funcionamiento del disparo, independientemente del modo, se basa en comparar constantemente cada señal con su umbral, de forma que cuando se cumpla la condición de disparo, se genera un flanco positivo que ejecuta el *Triggered Subsystem* conectado al primer puerto del bloque *merge*. Este subsistema, es idéntico al de la Fig. 3.14, por lo que al ejecutarse escribirá un 1 en la señal a la salida del bloque *merge*, indicando que comienza así la fase inspiratoria. En cuanto a los umbrales, el de flujo habitualmente es del orden de 0,05 L/s; el de presión se establece en unos 0,5 cmH<sub>2</sub>O por debajo del nivel de PEEP, de ahí que se resten la señal  $Paw$  y la variable  $PEEP$ . Ambas señales tienen su correspondiente bloque *Delay*, para evitar los bucles algebraicos.

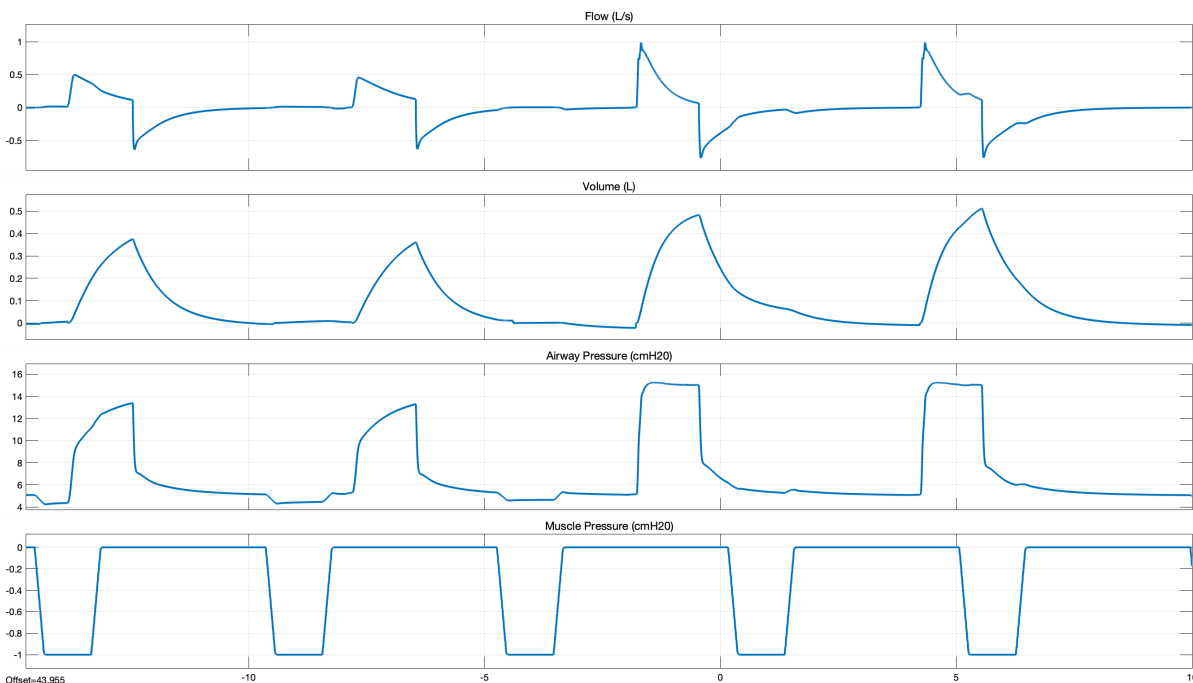


Fig. 3.18: Comparación de resultados de simulación del modo PC-CMV con control en lazo abierto (*primera y segunda respiración*) y en lazo cerrado (*tercera y cuarta respiración*) (FR = 10 resp/minuto;  $T_i:T_e = 0,25$ ;  $P_{control} = 15$  cmH<sub>2</sub>O;  $Pramp = 0,125$  s; PEEP = 5 cmH<sub>2</sub>O;  $R = 9$  cmH<sub>2</sub>O/(L/s);  $E = 20$  cmH<sub>2</sub>O/L;  $P = 5$ ;  $I = 70$ ;  $D = 0,1$ ) [elaboración propia]

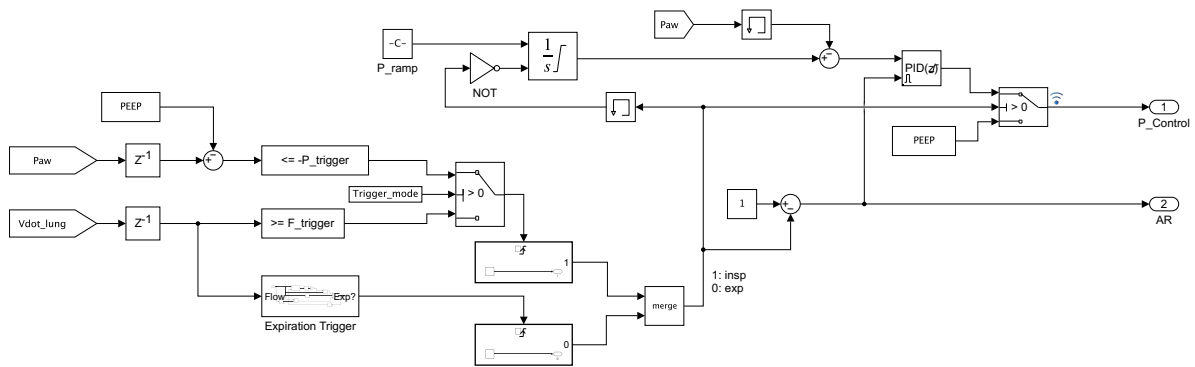


Fig. 3.19: Detalle del subsistema *Control* en el modelo que implementa el modo PSV [elaboración propia]

En cuanto al ciclado, se produce cuando el flujo desciende hasta una fracción del pico máximo alcanzado durante esa fase inspiratoria. Esto se detecta mediante el subsistema *Expiration Trigger* (Fig. 3.20), y cuando lo hace genera un flanco positivo que ejecuta el segundo *Triggered Subsystem*, el cual escribe un 0 en la señal de salida del *merge*, indicando el comienzo de la espiración.

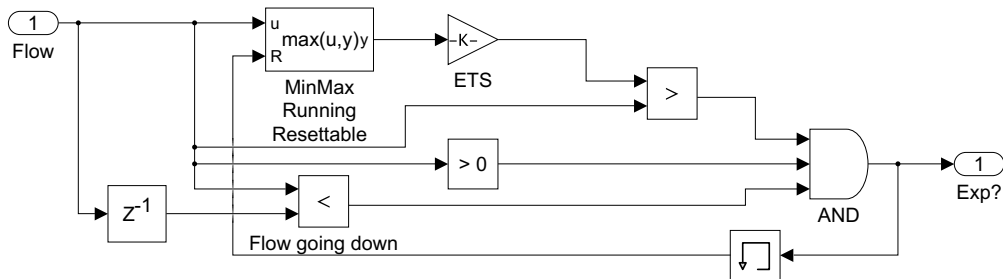


Fig. 3.20: Detalle de subsistema *Expiration Trigger* en el modelo que implementa el modo PSV [elaboración propia]

Para detectar el comienzo de la inspiración, se comprueban varias condiciones. La primera, que efectivamente se trata de una inspiración, por lo que el flujo es mayor que cero. La segunda, que el flujo ya ha comenzado a descender, por lo que se comprueba que el valor de flujo actual sea menor que en el instante anterior. Por último, se verifica que se ha alcanzado el nivel de flujo establecido para comenzar la espiración. Para conseguir esto último, se utiliza el bloque *MinMax Running Resettable*, el cual escribe a su salida el valor máximo (en este caso), de todos los valores pasados de su entrada  $u$ . Dicho valor máximo es reseteado cuando la señal del puerto  $R$  se evalúa a *true* (1). Una vez obtenido el valor máximo del flujo, se multiplica por el parámetro ETS (habitualmente establecido en un 25%), obteniéndose así el umbral de flujo que cicla la inspiración. Estas tres condiciones deben cumplirse simultáneamente, por lo que se integran las salidas de los tres comparadores en un bloque *AND*, que produce un 1 cuando se cumplen todos los requisitos. Esta señal sirve tanto para ejecutar el *Triggered Subsystem* del nivel superior como para resetear el bloque *MinMax Running Resettable*.

Del mismo modo que en los casos anteriores, la señal de apertura de la válvula espiratoria consiste en un 0 durante la inspiración y un 1 durante la espiración.

En la Fig. 3.21 se observan las curvas resultado de la simulación del modo PSV. En ellas se observa como, ahora sí, el esfuerzo del paciente es el que inicia la inspiración; además, se advierte como la fase espiratoria comienza cuando el flujo alcanza un cuarto de su valor de pico.

### 3.2.5. PAV

De nuevo, este modo de control de presión utiliza el mismo modelo de ventilador que los dos anteriores. Sin embargo, el esquema de control presenta ciertas diferencias (Fig. 3.22).

Se observa la misma estrategia basada en el elemento *Switch*. La lógica que rige la señal de control de la fase respiratoria se diseña en base al ventilador descrito en [22]. Cuando el valor del flujo medido desciende del umbral de 0,05 L/s, se abre la válvula espiratoria y se presuriza el circuito al nivel de PEEP. La inspiración, por contra, se inicia cuando se supera dicho umbral, momento en el que se cierra la válvula espiratoria y comienza a aplicarse la consigna de presión.

La consigna de presión se calcula en tiempo real en base a los valores instantáneos de flujo y volumen inspirados, multiplicándolos por unas ganancias en términos resistivos y elásticos, respectivamente. Estos parámetros, de acuerdo con los métodos de ajuste del modo PAV detallados en el apartado 2.1.4.3, se establecen multiplicando la elastancia y resistencia del paciente por las ganancias  $K1$  y  $K2$ , respectivamente, definidas en tanto por ciento (%). Para el volumen inspirado, se utiliza la señal  $V_{vent}$ , que se recuerda se reseteaba a cero al final de cada inspiración y durante toda la espiración. Del mismo modo, puesto que la espiración es pasiva y no tiene sentido calcular la señal de consigna durante esta fase, se satura la señal de flujo a un límite inferior de cero. A estas dos señales, multiplicadas por sus ganancias, se le añade un *offset* que no es más que el valor de la PEEP.

Igual que en los casos anteriores, se cierra el lazo con la señal de presión en la vía aérea y se implementa un controlador PID. En este caso, se conservan los parámetros de los otros modos menos el término proporcional, que se disminuye a 2, puesto que la consigna a seguir ahora no es constante y valores más grandes producirían inestabilidad.

En la Fig. 3.23 se observan distintos resultados fruto de la simulación de este modelo. En una primera respiración, con  $K1$  y  $K2 = 0,5$ , la asistencia ventilatoria amplifica la presión generada por el paciente

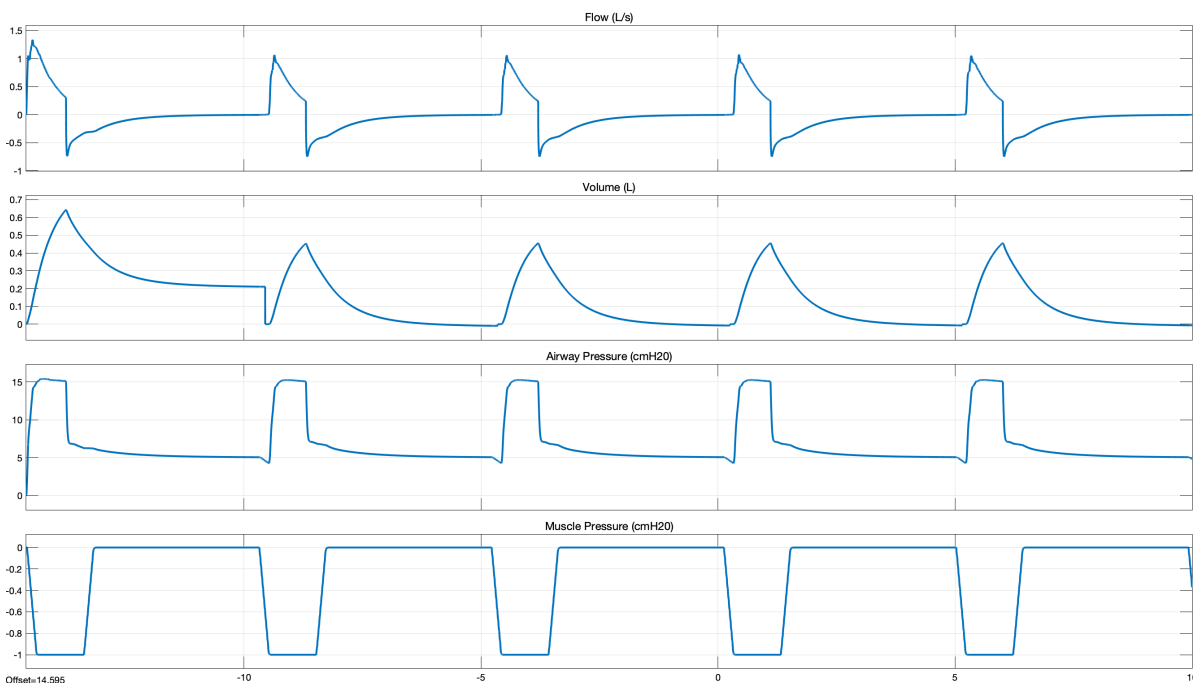


Fig. 3.21: Resultados de simulación del modo PSV (FR del paciente = 12 resp/minuto;  $P_{control} = 15$  cmH<sub>2</sub>O;  $Pramp = 0,125$  s; Umbral de disparo = 0,5 cmH<sub>2</sub>O; PEEP = 5 cmH<sub>2</sub>O;  $R = 9$  cmH<sub>2</sub>O/(L/s);  $E = 20$  cmH<sub>2</sub>O/L) [elaboración propia]



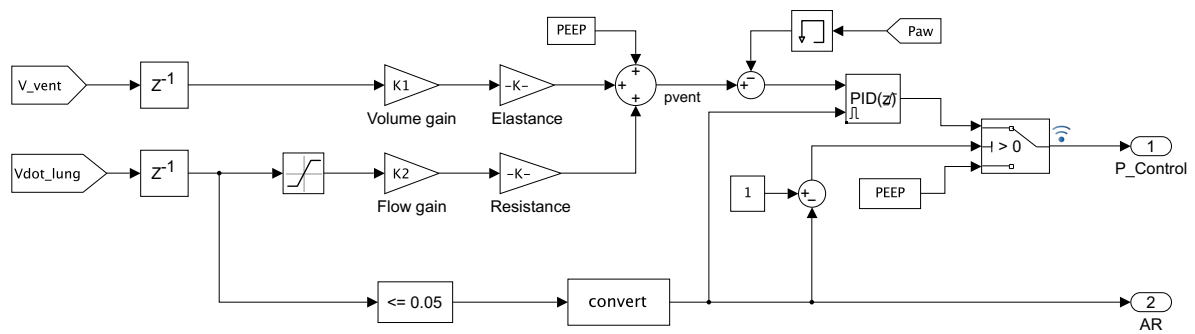


Fig. 3.22: Detalle del subsistema *Control* en el modelo que implementa el modo PAV [elaboración propia]

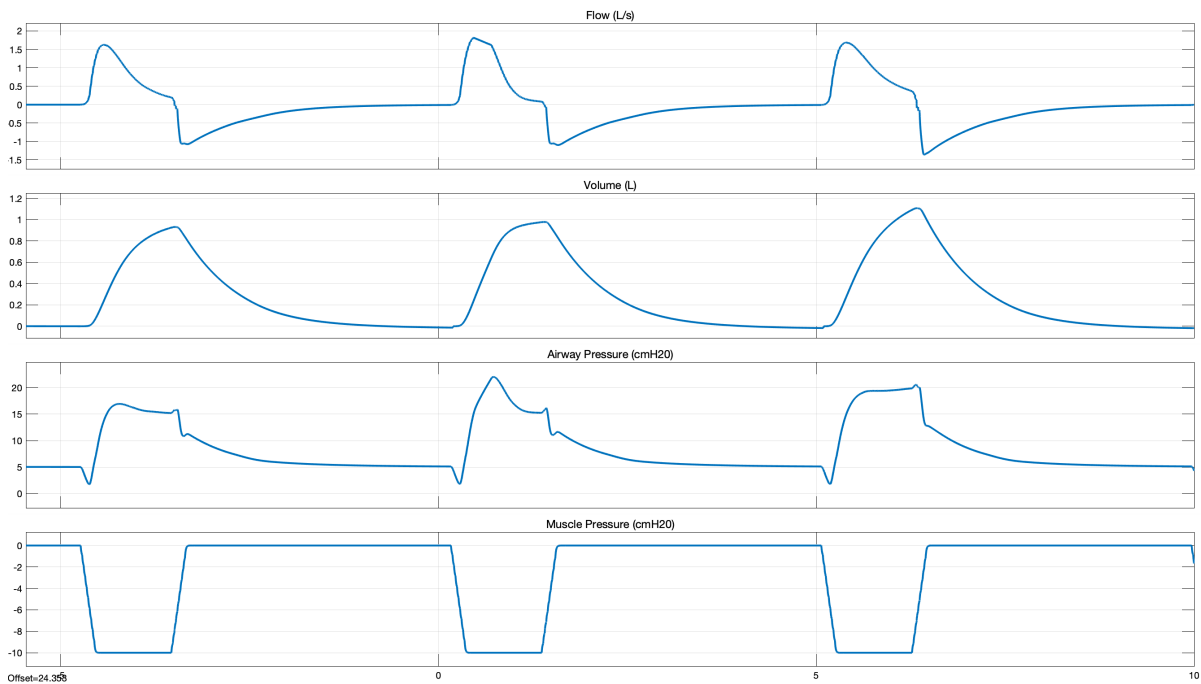


Fig. 3.23: Resultados de simulación del modo PAV ( $K1 = 0,5$  (1),  $0,5$  (2),  $0,6$  (3);  $K2 = 0,5$  (1),  $0,6$  (2),  $0,5$  (3);  $PEEP = 5$  cmH<sub>2</sub>O;  $R = 9$  cmH<sub>2</sub>O/(L/s);  $E = 20$  cmH<sub>2</sub>O/L) [elaboración propia]

en un mismo factor durante toda la inspiración. Esto se traduce, en este caso, en que la presión en la vía aérea es constante al igual que el esfuerzo inspiratorio (con una pequeña sobreoscilación fruto de la estrategia de control). En la segunda respiración se ha modificado únicamente el valor de  $K2$  a  $0,6$ , lo cual se refleja en una mayor presión al comienzo de la inspiración, cuando el flujo es mayor. Por último, en la tercera respiración la configuración fue  $K1 = 0,6$  y  $K2 = 0,5$ ; como es de esperar, la presión crece hacia el final de la fase inspiratoria, provocando que aumente el volumen tidal.

## 3.3. INTERFAZ GRÁFICA

### 3.3.1. *Front-end*

La interfaz gráfica para la interacción con el modelo se desarrolla en el entorno App Designer de MATLAB. La aplicación, denominada *VentApp*, consta de 3 paneles principales:

- **Patient.** En esta pantalla se ajustan los parámetros del paciente tales como resistencia o elastancia, así como la forma, magnitud y frecuencia del esfuerzo inspiratorio (Fig. 3.24). Se dispone de un menú desplegable con tres perfiles predefinidos de la presión muscular: *Passive* —el esfuerzo es nulo, para modos CMV—, *Simple Trigger* —breve esfuerzo inspiratorio, pensado para usar con el modo PSV— y *Trapezoidal Profile* —configurable a medida con esfuerzo tanto inspiratorio como espiratorio, pensado para el modo PAV—.

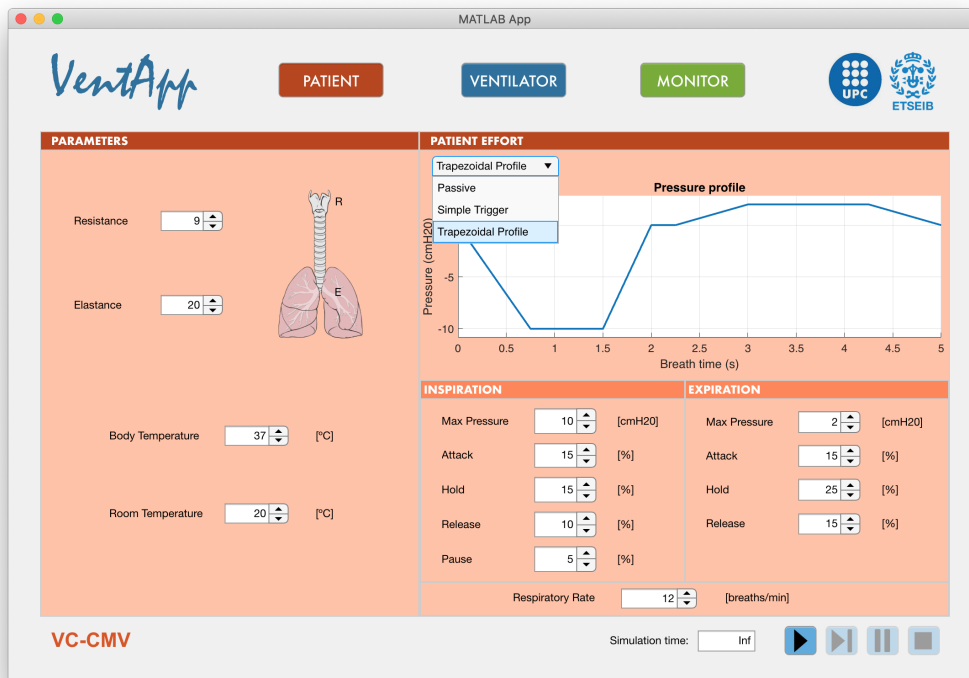


Fig. 3.24: Panel *Patient* en la aplicación *VentApp* [elaboración propia]

- **Ventilator.** Esta ventana permite escoger entre los cuatro modos de ventilación disponibles y ajustar los parámetros que intervienen en cada uno de ellos. Además, se incorpora una breve descripción de cada modo ventilatorio de manera informativa. Los parámetros se encuentran agrupados de forma que se percibe claramente cuales son comunes a qué modos. A mayores, los modos VC-CMV, PC-CMV y PSV incorporan una gráfica que muestra la señal de consigna configurada, actualizándose automáticamente con cada modificación de los parámetros que en ella intervienen. Las Fig. 3.25 a 3.28 muestran el panel *Ventilator* para los cuatro modos de ventilación.
- **Monitor.** En esta pantalla se muestran en tiempo real las señales de flujo, volumen, presión en la vía aérea y presión muscular, fruto de la simulación en curso. Para mejor visualización de la interacción ventilador-paciente, las dos señales de presión se muestran en la misma gráfica (Fig. 3.29).

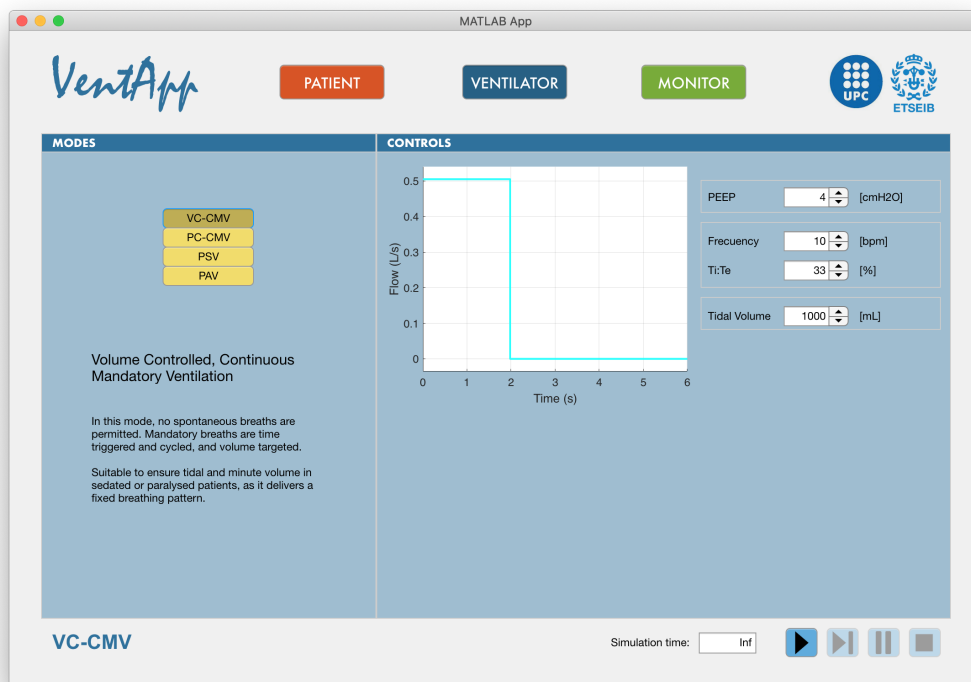


Fig. 3.25: Panel *Ventilator* en modo VC-CMV en la aplicación VentApp [elaboración propia]

Además, en la ventana general de la aplicación, accesible desde todos los paneles, se encuentran los controles para la ejecución del modelo. Se cuenta con un campo editable para establecer el tiempo de simulación en segundos, y los botones habituales en el entorno de Simulink, que se habilitan y deshabilitan en función del estado de la simulación: *start*, *next step*, *pause* y *stop*. También accesible en todo momento, una etiqueta de texto indica el modo en el que se encuentra el ventilador en cada instante.

### 3.3.2. *Back-end*

El funcionamiento interno de la aplicación, y su interacción con el modelo, se consigue en parte mediante el acceso a las variables del espacio de trabajo base, que son las mismas que utiliza Simulink. La simulación del modelo se da, por comando, en un segundo plano, sin ejecutar el entorno gráfico de Simulink. La monitorización se consigue mediante el acceso a unos ficheros que se graban en tiempo real durante la simulación. A continuación, se detallan estos conceptos en más profundidad.

#### 3.3.2.1. Modificación del modelo

Como se introdujo en el apartado 2.2.2, al arrastrar un componente de la librería en el *lienzo*, App Designer añade a las propiedades públicas de la clase que define la aplicación un objeto del tipo correspondiente al componente añadido. Además, define dentro del método privado que crea los componentes de la *app* las propiedades con las que se inicializa el objeto. Otras de las funcionalidades del entorno App Designer es que, haciendo click derecho sobre un componente, se puede crear automáticamente o acceder a sus *callbacks* —métodos privados en los que se define la acción a realizar

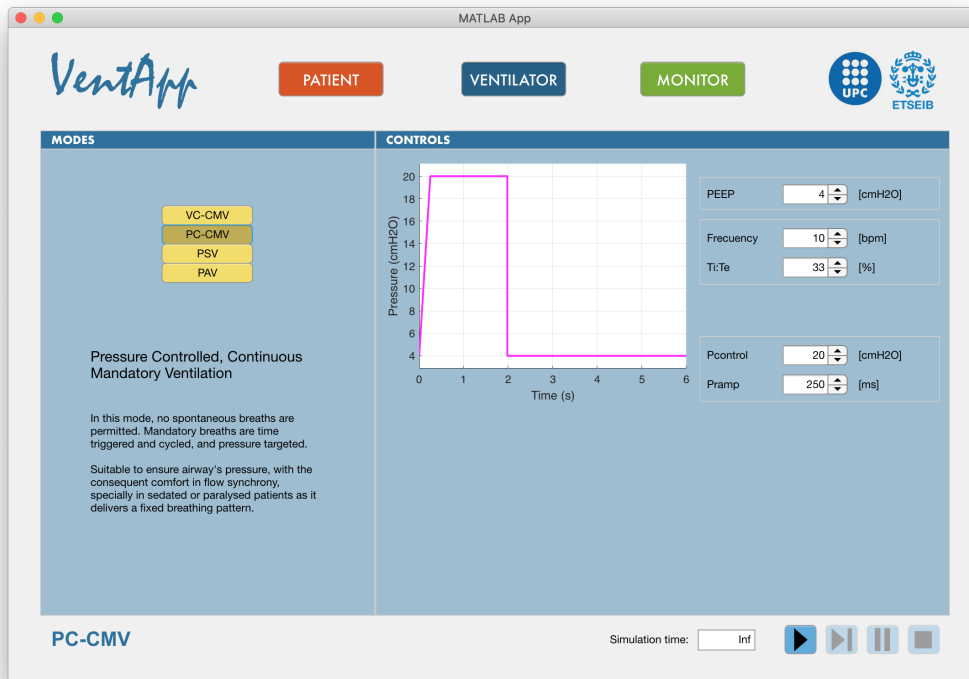


Fig. 3.26: Panel *Ventilator* en modo PC-CMV en la aplicación VentApp [elaboración propia]

cuando se interactúa con el componente, en caso de que este admita interacción—. Esta es la base de la interacción entre los componentes gráficos y las entrañas de la aplicación.

A modo de ejemplo, se incluye la función *callback* del componente que establece el valor de elastancia del paciente (Listado 3.5), que se ejecuta mientras su valor está cambiando.

Listado 3.5: *Callback* del componente que establece la elastancia del paciente

```

1  % Value changing function: ElastanceSpinner
2  function ElastanceSpinnerValueChanging(app, event)
3      changingValue = event.Value;
4      assignin('base', 'E_respiratory', changingValue);
5      resize = changingValue/app.ElastanceSpinner.Limits(2);
6      app.img_lung.Position(3) = int16(100*(1-(resize/4)));
7      app.img_lung.Position(4) = int16(100*(1-(resize/4)));
8      app.img_lung.Position(1) = 70 - (app.img_lung.Position(3)/2);
9      app.img_lung.Position(2) = 82 - (app.img_lung.Position(4)/2);
10     app.VG_result_Label.Text = sprintf('%.2f',...
11         changingValue*evalin('base', 'K1'));
12 end

```

En esta función se observan diferentes acciones. Primero, se asigna el valor actual del componente a la variable del espacio de trabajo *base E\_respiratory*, mediante las dos primeras instrucciones. Además, se calcula la relación entre el valor actual establecido y el máximo para este elemento (fijado en 50 cmH<sub>2</sub>O/L), y con esta ratio se redefine el tamaño de la imagen de los pulmones así como su posición en el lienzo, para mantenerla en el mismo punto (líneas 5 a 9). De esta forma, cuanto mayor sea la elastancia, menor será el tamaño de la imagen del pulmón, ilustrando así de una manera un tanto bruta pero didáctica el efecto de este parámetro. Del mismo modo, al aumentar el valor de resistencia, la imagen de la tráquea se estrechece. La última instrucción modifica la etiqueta de texto disponible en la

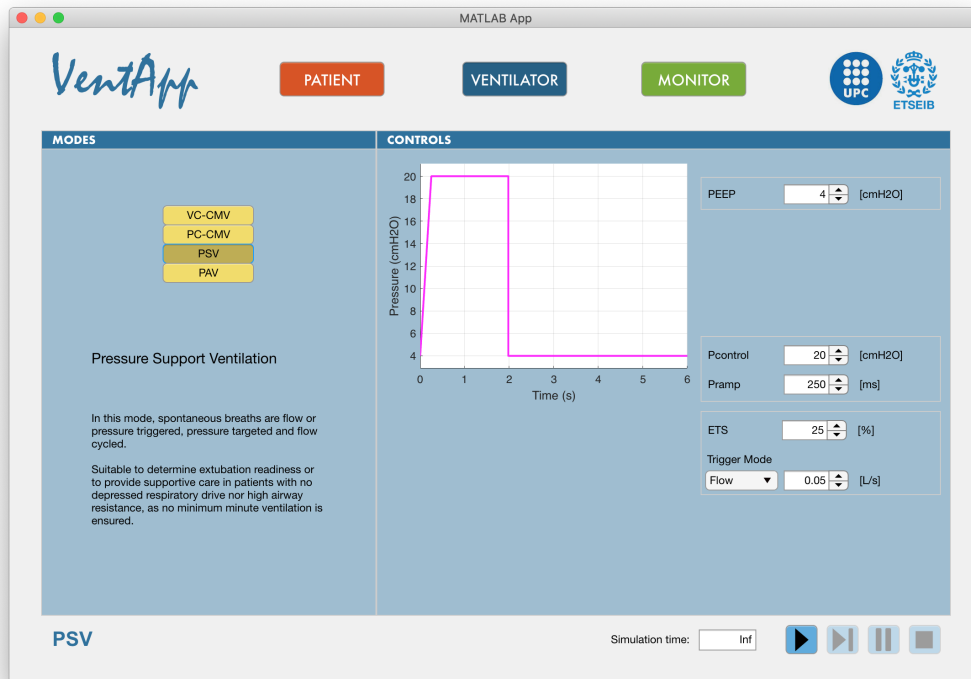


Fig. 3.27: Panel *Ventilator* en modo PSV en la aplicación VentApp [elaboración propia]

ventana *Ventilator* en el modo PAV, que indica el valor de ganancia configurado en base a los valores de elastancia y  $K1$  (misma funcionalidad para la componente resistiva).

De un modo similar funcionan los *spinner* del panel *Patient Effort*. En este caso, además, llaman a un método privado que actualiza el gráfico *Pressure Profile* con la configuración de parámetros establecida (*plot\_pmus* en el Listado 3.6). Por otra parte, cuando uno de estos parámetros es modificado, habilita un botón de aplicación de cambios, que a su vez solo se vuelve visible cuando la simulación está en marcha; de este modo, se pueden actualizar algunos parámetros del modelo durante la simulación, para mayor funcionalidad en la configuración de estos valores. Concretamente, los parámetros que se pueden modificar durante la simulación son aquellos que no alteran los componentes de Simscape, pues para que el cambio surja efecto en estos es necesario recompilar el modelo. En la ventana *Ventilator* se añade otro botón de aplicación de cambios, con la misma dinámica de funcionamiento pero para los parámetros del ventilador ajustables desde esa ventana. En total, los parámetros excluidos de la actualización en tiempo real son: la resistencia y elastancia del paciente, las temperaturas del cuerpo y de la habitación, el nivel de PEEP, el tiempo de simulación y, naturalmente, el modo de ventilación. Los componentes que controlan estos parámetros, aunque visible, se muestran deshabilitados durante la simulación.

Listado 3.6: *Callback* del componente que establece la presión inspiratoria del paciente, función que actualiza el gráfico *Pressure Profile* y *callback* de los botones de aplicación de cambios en tiempo real

```

1 % Value changed function: Insp_PressureSpinner
2 function Insp_PressureSpinnerValueChanged(app, event)
3     value = app.Insp_PressureSpinner.Value;
4     assignin('base', 'insp_P', value);
5     app.plot_pmus;
6     app.APPLYCHANGESButton.Enable = 'on';
7 end
8

```

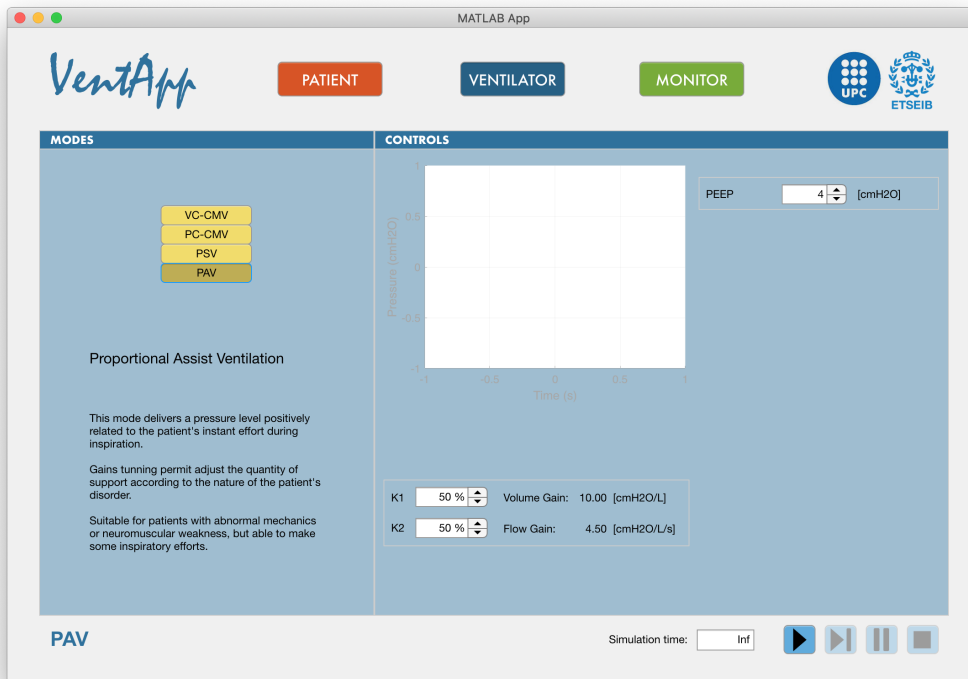


Fig. 3.28: Panel *Ventilator* en modo PAV en la aplicación VentApp [elaboración propia]

```

9 function plot_pmus(app)
10     insp_P = evalin('base','insp_P');
11     insp_increase = evalin('base','insp_increase');
12     insp_hold = evalin('base','insp_hold');
13     insp_release = evalin('base','insp_release');
14     insp_pause = evalin('base','insp_pause');
15     exp_P = evalin('base','exp_P');
16     exp_increase = evalin('base','exp_increase');
17     exp_hold = evalin('base','exp_hold');
18     exp_release = evalin('base','exp_release');
19     pat_RR = evalin('base','pat_RR');
20     if (insp_increase+insp_hold+insp_release+insp_pause+...
21         exp_increase+exp_hold+exp_release) ~= 1
22         app.PercentageError.Visible = true;
23     else
24         app.PercentageError.Visible = false;
25         t = [0, insp_increase*60/pat_RR, (insp_increase+insp_hold)*60/pat_RR,...
26             (insp_increase+insp_hold+insp_release)*60/pat_RR, ...
27             (insp_increase+insp_hold+insp_release+insp_pause)*60/pat_RR, ...
28             (insp_increase+insp_hold+insp_release+insp_pause+exp_increase)...
29             *60/pat_RR, (insp_increase+insp_hold+insp_release+insp_pause+...
30             exp_increase+exp_hold)*60/pat_RR, (insp_increase+insp_hold+...
31             insp_release+insp_pause+exp_increase+exp_hold+exp_release)*...
32             60/pat_RR].';
33         values = [0 -insp_P -insp_P 0 0 exp_P exp_P 0].';
34         plot(app.PatientPressureAxes,t,values, 'LineWidth',2);
35     end
36 end
37
38 % Button pushed function: APPLYCHANGESButton,
39 % APPLYCHANGESButton_2
40 function APPLYCHANGESButtonPushed(app, event)

```



Fig. 3.29: Panel *Monitor* en la aplicación *VentApp* con los resultados de una simulación en modo PSV [elaboración propia]

```

41     set_param(app.mdl, 'SimulationCommand', 'update');
42     app.APPLYCHANGESButton.Enable = 'off';
43     app.APPLYCHANGESButton_2.Enable = 'off';
44 end

```

El funcionamiento del método `plot_pmus` es almacenar en unas variables locales el valor de las variables del espacio de trabajo base que definen la  $P_{mus}$  en el modelo. A partir de ahí, evalúa la condición de que los porcentajes de tiempos deben sumar un 100 %, en cuyo caso grafica la curva correspondiente; en caso contrario, muestra un mensaje de error. La *callback* de los botones de aplicación de cambios actualizan el modelo para luego volver a inhabilitar dichos botones, a la espera de modificar de nuevo algún parámetro.

Para implementar el menú desplegable con los ajustes predefinidos de esfuerzo del paciente, se implementa una instrucción *switch case* que, según la opción seleccionada del menú, actualiza el valor de los *spinners* con los valores predefinidos en cada caso. Fuera de esta instrucción, se actualizan las variables del espacio base en función del valor de estos *spinners*, se llama a la función `plot_pmus` y se habilita el botón de aplicación de cambios.

Mencionar que para cambiar entre los paneles *Patient*, *Ventilator* y *Monitor*, se utiliza un *Button Group*. Este elemento implementa una serie de botones (en este caso tres, en la parte superior) excluyentes entre sí; es decir, solo uno de ellos puede estar seleccionado a la vez. La *callback* de este elemento se muestra en el Listado 3.7. La dinámica consiste en ocultar los tres paneles para posteriormente, mediante una sentencia *switch case*, mostrar el correspondiente. En esta instrucción también se modifica el color de la etiqueta de texto que indica el modo ventilatorio seleccionado, con fines puramente estéticos.

Listado 3.7: *Callback* del *button group* que permite cambiar la visualización entre los paneles *Patient*, *Ventilator* y *Monitor*

```

1  % Selection changed function: MainButtons
2  function MainButtonsSelectionChanged(app, event)
3      selectedButton = app.MainButtons.SelectedObject;
4      app.Patient.Visible = 'off';
5      app.Ventilator.Visible = 'off';
6      app.Monitor.Visible = 'off';
7      switch selectedButton.Text
8          case 'PATIENT'
9              app.Patient.Visible = 'on';
10             app.ModeLabel.FontColor = '0.85,0.33,0.10';
11         case 'VENTILATOR'
12             app.Ventilator.Visible = 'on';
13             app.ModeLabel.FontColor = '0.17,0.44,0.62';
14         case 'MONITOR'
15             app.Monitor.Visible = 'on';
16             app.ModeLabel.FontColor = '0.36,0.54,0.13';
17     end
18 end

```

Para la selección del modo ventilatorio se utiliza también un *button group*, con cuatro botones en este caso. En el Listado 3.8 se muestran las instrucciones asociadas a la *callback* de este componente. Primero, se actualiza la visibilidad de los textos explicativos del mismo modo que se realiza en el caso anterior. Dentro de la sentencia *switch* también se actualiza la variable del espacio base *mode*, que es la que sirve para alternar entre los distintos modos ventilatorios. Además, se actualizan la etiqueta identificadora del modo seleccionado y la visibilidad de los paneles en los que se agrupan los parámetros de configuración de los distintos modos. En total se dispone de seis paneles de parámetros: uno para la PEEP, uno para los modos con secuencia CMV, dos para el control de volumen y de presión (PAV no inclusive), respectivamente, uno para la sincronización en el PSV y, por último, uno para el modo PAV. Dentro del *switch* también se actualiza la gráfica presente, para lo cual se crean dos funciones: *plot\_VC* y *plot\_PC*. En el modo PAV, que no dispone de parámetros graficables, se simula una deshabilitación de la gráfica.

Listado 3.8: *Callback* del *button group* que permite seleccionar los distintos modos de ventilación

```

1  % Selection changed function: ButtonGroup_Modes
2  function ButtonGroup_ModesSelectionChanged(app, event)
3      selectedButton = app.ButtonGroup_Modes.SelectedObject;
4      app.VCCMV_Explan_Label.Visible = 'off';
5      app.VCCMV_Explan_Label_2.Visible = 'off';
6      app.PAV_Explan_Label_2.Visible = 'off';
7      app.PAV_Explan_Label.Visible = 'off';
8      app.PCCMV_Explan_Label_2.Visible = 'off';
9      app.PCCMV_Explan_Label.Visible = 'off';
10     app.PSV_Explan_Label_2.Visible = 'off';
11     app.PSV_Explan_Label.Visible = 'off';
12     app.Panel_VC.Visible = 'off';
13     app.Panel_CMV.Visible = 'off';
14     app.Panel_PC.Visible = 'off';
15     app.Panel_PS.Visible = 'off';
16     app.Panel_PAV.Visible = 'off';
17     app.VentilatorAxes.XColor = '0.15,0.15,0.15';
18     app.VentilatorAxes.YColor = '0.15,0.15,0.15';
19     switch selectedButton.Text
20         case 'VC-CMV'
21             app.VCCMV_Explan_Label.Visible = 'on';
22             app.VCCMV_Explan_Label_2.Visible = 'on';
23             assignin('base','mode',1);
24             app.ModeLabel.Text = ' VC-CMV';
25             app.Panel_VC.Visible = 'on';
26             app.Panel_CMV.Visible = 'on';
27             plot_VC(app);
28         case 'PC-CMV'
29             app.PCCMV_Explan_Label_2.Visible = 'on';
30             app.PCCMV_Explan_Label.Visible = 'on';
31             assignin('base','mode',2);
32             app.ModeLabel.Text = ' PC-CMV';

```



```

33         app.Panel_CMV.Visible = 'on';
34         app.Panel_PC.Visible = 'on';
35         plot_PC(app);
36     case 'PSV'
37         app.PSV_Explan_Label_2.Visible = 'on';
38         app.PSV_Explan_Label.Visible = 'on';
39         assignin('base','mode',3);
40         app.ModeLabel.Text = ' PSV';
41         app.Panel_PC.Visible = 'on';
42         app.Panel_PS.Visible = 'on';
43         plot_PC(app);
44     case 'PAV'
45         app.PAV_Explan_Label_2.Visible = 'on';
46         app.PAV_Explan_Label.Visible = 'on';
47         assignin('base','mode',4);
48         app.ModeLabel.Text = ' PAV';
49         app.Panel_PAV.Visible = 'on';
50         app.VentilatorAxes.XColor = '0.66,0.66,0.66';
51         app.VentilatorAxes.YColor = '0.66,0.66,0.66';
52         plot(app.VentilatorAxes,0,0);
53     end
54 end
55
56
57 function plot_VC(app)
58     app.VentilatorAxes.YLabel.String = "Flow (L/s)";
59     RR = evalin('base','respiratory_rate');
60     IE = evalin('base','IE_ratio');
61     VT = evalin('base','tidal_volume');
62     tinsp = IE*60/RR;
63     t = [0, tinsp, tinsp+1e-4, 60/RR];
64     flow = VT/tinsp;
65     assignin('base','flow',flow);
66     values = [flow,flow,0,0];
67     plot(app.VentilatorAxes, t, values, 'c', 'LineWidth',2);
68 end
69
70 function plot_PC(app)
71     app.VentilatorAxes.YLabel.String = "Pressure (cmH2O)";
72     RR = evalin('base','respiratory_rate');
73     IE = evalin('base','IE_ratio');
74     Peep = evalin('base','PEEP');
75     Pc = evalin('base','Pcontrol');
76     Pr = evalin('base','Pramp');
77     tinsp = IE*60/RR;
78     t = [0, Pr, tinsp, tinsp+1e-4, 60/RR];
79     values = [Peep,Pc,Pc,Peep,Peep];
80     plot(app.VentilatorAxes, t, values, 'm', 'LineWidth',2);
81 end

```

Las funciones *plot\_VC* y *plot\_PC* son muy similares. Ambas comienzan actualizando la etiqueta del objeto *VentilatorAxes* (el gráfico), para mostrar la magnitud adecuada. Posteriormente, obtienen el valor de las variables relevantes del espacio de trabajo base, y actualizan el *VentilatorAxes* con la curva de consigna calculada. En la función *plot\_VC*, además, se calcula el flujo inspiratorio a partir de (2.6) y se graba de vuelta el valor en la variable *flow* en el espacio base. Los componentes cuyos parámetros asociados modifican algún valor de los que intervienen en estas consignas, llaman también en sus *callbacks* a la función que corresponda de estas dos. En caso de parámetros como la frecuencia respiratoria, se comprueba el modo seleccionado para llamar a una u otra función.

### 3.3.2.2. Simulación y monitorización

Al iniciar *VentApp*, se ejecuta automáticamente una función de inicialización (*startupFcn*) en la que se disponen de varias sentencias. Entre ellas se encuentra la llamada a la función de MATLAB

*prj\_config\_fcn2.m*. Esta función, además de organizar las carpetas donde Simulink genera los datos temporales, permite inicializar o cerrar el modelo mediante el argumento *'set\_up'* o *'clean\_up'*. En el primer caso, se ejecuta la instrucción `load_system mdl;`, donde *mdl* almacena el nombre del modelo *Simulator*, lo cual lo inicializa en segundo plano.

Una vez se ha cargado el modelo, ya se puede interactuar con la aplicación y comenzar las simulaciones. Para ello se dispone de una serie de botones que implementan los controles habituales en la ejecución de una simulación. A modo de ejemplo, se muestra en el Listado 3.9 el detalle de la *callback* del botón *start*.

Listado 3.9: *Callback* del botón *start* que inicia la simulación del modelo

```

1 % Button pushed function: START
2 function STARTButtonPushed(app, event)
3     app.START.Enable = 'off';
4     app.StatusLabel.Text = 'initializing';
5     pause(0.2);
6     set_param(app.mdl, 'SimulationCommand', 'Start');
7     axis ([app.FlowAxes, app.VolumeAxes, app.PawAxes], ...
8         [0, min(evalin('base', 'tsim'), 20), -Inf, Inf]);
9     start(app.plot_timer);
10    app.NEXTSTEP.Enable = 'off';
11    app.STOP.Enable = 'on';
12    app.PAUSE.Enable = 'on';
13    app.started = true;
14    enable_pams(app, 'off');
15    app.APPLYCHANGESButton.Visible = 'on';
16    app.APPLYCHANGESButton_2.Visible = 'on';
17 end

```

En un primer momento, se deshabilita el propio botón y se muestra un texto con el mensaje *initializing*, para indicar que la compilación ya ha comenzado. La pausa de 0,2 segundos se establece para asegurar la ejecución de estas dos instrucciones antes de que la siguiente, que es la propia que inicializa la simulación, retenga la actualización de estos dos componentes. La siguiente instrucción (líneas 7 y 8), establece la escala del eje temporal de las tres gráficas incluidas en la ventana *Monitor* (*FlowAxes*, *VolumeAxes* y *PawAxes*). Si el tiempo de simulación establecido es menor que 20, el eje X se ajusta entre 0 y ese tiempo; en caso contrario, se establece en 20 segundos.

La siguiente instrucción inicia el temporizador *plot\_timer*, del cual se hablará a continuación. Las líneas 10 a 12 habilitan y deshabilitan los demás botones de control de la simulación, mientras que la propiedad *started* se utiliza para tareas de comprobación y ejecución condicionada. El método *enable\_pams*, ejecutado con el argumento *'off'*, deshabilita aquellos controles que no son modificables en tiempo real. Por último, se muestran los botones para la aplicación de cambios durante la simulación, aunque deshabilitados hasta que no se modifique alguno de los parámetros.

## Temporizador

El temporizador (*plot\_timer*) es un objeto que se crea en la función *startupFcn*, después de la actualización de todas las variables del espacio base conforme a los valores de los componentes y de la inicialización de todas las gráficas de la aplicación. Este temporizador es un elemento que permite programar la ejecución de ciertas instrucciones una o múltiples veces, en base al tiempo transcurrido. La creación de este objeto se hace tal como se muestra en el Listado 3.10.

Listado 3.10: Inicialización del objeto temporizador

```

1 app.plot_timer = timer('Period', app.Ts, ...
2     'ExecutionMode', 'fixedSpacing', ...
3     'TasksToExecute', Inf, 'BusyMode', 'drop');
4 app.plot_timer.TimerFcn = @(~,~) app.timerfcn;

```

En este caso, las instrucciones a ejecutar son definidas mediante la *callback timerfcn*. El tiempo de retraso entre las llamadas a esta función se establece mediante el argumento *Period* con un valor definido por la propiedad *Ts*. El modo de ejecución es *fixedSpacing*, lo cual quiere decir que el tiempo transcurrido antes de realizar una nueva llamada a *timerfcn* se contabiliza desde el final de la anterior ejecución. Mediante el argumento *TaskToExecute* definido a *Inf* se configura el *timer* para que llame a *timerfcn* indefinidamente, mientras esté en marcha. El parámetro *BusyMode* a *drop* establece que, en caso de colas en la ejecución de la *callback* del temporizador, la regla es descartar las llamadas anteriores.

El objeto de este temporizador es definir en su *callback* las instrucciones que actualizan las gráficas del panel *Monitor* en tiempo real, pero sin bloquear la ejecución del simulador, lo cual ocurriría si se actualizasen mediante una función en constante ejecución.

Para ello, primero es necesario guardar los resultados de la simulación en tiempo real. MATLAB/Simulink permite esto tan solo mediante la grabación de los datos en ficheros *.mat*, de ahí los cuatro bloques que se visualizan en la Fig. 3.1, y que almacenan los datos en sendos ficheros MAT en la ruta *VentApp\_data/sim\_data/*. Para acceder a estos datos, es necesario referirse a la función *init\_plots* (Listado 3.11), llamada durante la ejecución de *startupFcn*.

Listado 3.11: Función que inicializa las gráficas del panel *Monitor*

```

1 function init_plots(app)
2     app.pf = plot(app.FlowAxes,0,0, 'LineWidth',2);
3     app.pf.XDataSource = 'flow_sim(1,:)';
4     app.pf.YDataSource = 'flow_sim(2,:)';
5     app.pv = plot(app.VolumeAxes,0,0, 'LineWidth',2);
6     app.pv.XDataSource = 'vol_sim(1,:)';
7     app.pv.YDataSource = 'vol_sim(2,:)';
8     app.ppaw = plot(app.PawAxes,0,0,0,0,'LineWidth',2);
9     app.ppaw(1,1).XDataSource = 'paw_sim(1,:)';
10    app.ppaw(1,1).YDataSource = 'paw_sim(2,:)';
11    app.ppaw(2,1).XDataSource = 'pmus_sim(1,:)';
12    app.ppaw(2,1).YDataSource = 'pmus_sim(2,:)';
13    axis ([app.FlowAxes,app.VolumeAxes,app.PawAxes],...
14          [0,min(evalin('base','tsim'),20),-Inf,Inf]);
15 end

```

Esta función, además de inicializar las gráficas a cero, configurar el grosor de las líneas y, nuevamente, el rango inicial del eje X, establece el origen de los datos de cada curva. La sentencia *plot* devuelve un puntero por cada objeto de línea que se crea en el gráfico correspondiente (véase *FlowAxes*, *VolumeAxes* y *PawAxes* (en este caso dos punteros)). Estos objetos de línea disponen de las propiedades *XDataSource* e *YDataSource*, del tipo *string* o cadena de texto. Especificando en este campo el nombre de una variable, se asocia con esta el origen de los datos de la curva para X e Y, respectivamente. En este caso, se asocia el origen de los datos a las variables *flow\_sim*, *vol\_sim*, *paw\_sim* y *pmus\_sim*, que precisamente son las variables que se guardan en los ficheros *flow.mat*, *volume.mat*, *paw.mat* y *pmus.mat* (Fig. 3.1).

En cuanto a la *timerfcn*, se adjunta su definición (Listado 3.12). Primeramente, se obtiene el estado de la simulación y se escribe en la etiqueta que indica esta condición. Además, se implementa una instrucción *switch case* en base a este estado; en caso de que la simulación se encuentre corriendo, se ejecuta la función *update\_plots*; en caso de que la simulación haya sido detenida, pero no se haya actualizado la variable *started*, quiere decir que la simulación ha llegado a su fin, por lo que es necesario ejecutar las instrucciones comprendidas en la *callback* del botón de *stop*. Tanto esta última como la función *update\_plots* se adjuntan también en el Listado 3.12

Listado 3.12: *Callback* del temporizador, función que actualiza las gráficas y *callback* del botón *stop*

```

1 function timerfcn(app)
2     app.StatusLabel.Text = get_param(app.mdl, ...
3     'SimulationStatus');
4     switch app.StatusLabel.Text
5     case 'running'
6         app.update_plots;

```

```

7         case 'stopped'
8             if (app.started)
9                 app.STOPButtonPushed;
10            end
11        end
12    end
13
14    function update_plots(app)
15        dp = './VentApp_data/sim_data/';
16        evalin('base', ['load ', dp, 'flow.mat flow_sim']);
17        if (numel(evalin('base', 'flow_sim'))>2)
18            refreshdata(app.FlowAxes)
19            evalin('base', ['load ', dp, 'volume.mat vol_sim']);
20            refreshdata(app.VolumeAxes)
21            evalin('base', ['load ', dp, 'paw.mat paw_sim']);
22            evalin('base', ['load ', dp, 'pmus.mat pmus_sim']);
23            refreshdata(app.PawAxes)
24            if (app.pf.XData(end) > 20)
25                axis ([app.FlowAxes, app.VolumeAxes, ...
26                    app.PawAxes], ...
27                    [app.pf.XData(end)-20, ...
28                    app.pf.XData(end), -Inf, Inf]);
29            end
30            drawnow
31        end
32    end
33
34    % Button pushed function: STOP
35    function STOPButtonPushed(app, event)
36        app.STOP.Enable = 'off';
37        set_param(app.mdl, 'SimulationCommand', 'stop');
38        stop(app.plot_timer);
39        app.START.Enable = 'on';
40        app.START.Visible = 'on';
41        app.CONTINUE.Visible = 'off';
42        app.CONTINUE.Enable = 'off';
43        app.PAUSE.Enable = 'off';
44        app.started = false;
45        pause(0.5);
46        update_plots(app);
47        app.StatusLabel.Text = get_param(app.mdl, ...
48            'SimulationStatus');
49        enable_pams(app, 'on');
50        app.APPLYCHANGESButton.Visible = 'off';
51        app.APPLYCHANGESButton.Enable = 'off';
52        app.APPLYCHANGESButton_2.Visible = 'off';
53        app.APPLYCHANGESButton_2.Enable = 'off';
54    end

```

Mediante las instrucciones de las líneas 16, 19, 21 y 22 se carga en el espacio de trabajo base los ficheros MAT generados durante la simulación. Esto crea (o actualiza) las variables que contienen estos ficheros —que son las establecidas como orígenes de datos de las curvas— con los datos obtenidos hasta el instante de la llamada a estas funciones. En un primer momento, se realiza tan solo con la variable *flow\_sim*, y tras comprobarse que existen datos que graficar, se realiza con las demás variables. Entre medias se llama a la función *refreshdata*, la cual actualiza los datos de las curvas en la gráfica pasada por parámetro. Posteriormente, se comprueba si el tiempo simulado hasta el momento es mayor de 20, en cuyo caso se actualizan los ejes horizontales de las gráficas para que muestren siempre los últimos 20 segundos de simulación, creando un efecto más animado. Por último, la sentencia *drawnow* actualiza las gráficas con los nuevos datos de las curvas.

Diseñada la aplicación, se empaqueta junto con los modelos de Simulink y demás recursos necesarios, y se crea el archivo *VentApp.mlappinstall*. Mediante su ejecución desde el entorno MATLAB, se procede a la instalación de la aplicación. Desde ese momento, *VentApp* es accesible desde el menú *APPS* de

la ventana principal de MATLAB (Fig. 3.30). Para abrir el modelo en Simulink, tan solo es necesario ejecutar la sentencia *Simulador*.

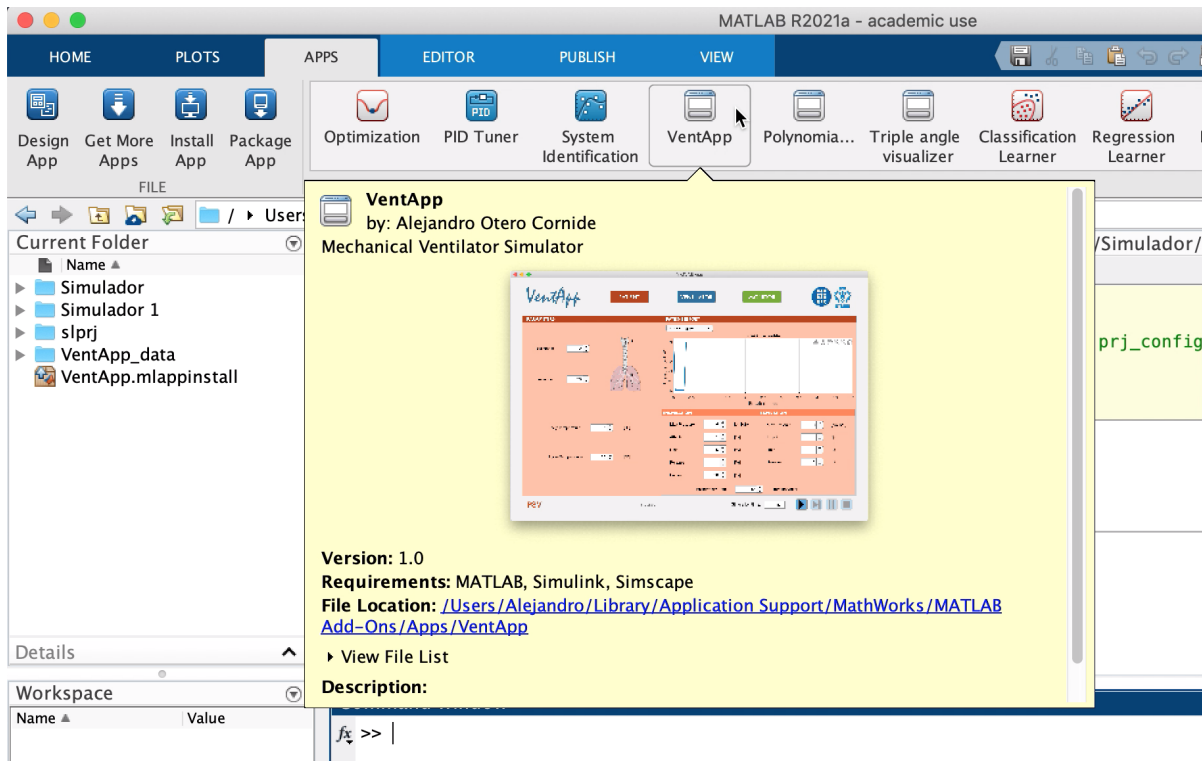


Fig. 3.30: Instalación de *VentApp* en el entorno MATLAB [elaboración propia]



# Validación

En este capítulo se detalla el proceso por el cual se legitima el correcto funcionamiento de la solución desarrollada. Se diferencian dos métodos para probar la valía del trabajo, uno cualitativo basado en literatura disponible para los modos de ventilación VC-CMV, PC-CMV y PAV; y otro cuantitativo para el modo PSV, basado en la comparación de resultados obtenidos mediante simulación y mediante un ensayo clínico real.

## 4.1. VALIDACIÓN CUALITATIVA

### 4.1.1. Modos CMV

Las curvas ideales esperadas para los modos de ventilación VC-CMV y PC-CMV son las mostradas en la Fig. 2.7, curvas B y A respectivamente. Sin embargo, estas señales se dibujan exentas de ruido o perturbaciones, como puede ser el esfuerzo inspiratorio del paciente. En la Fig. 4.1 se muestra un caso real de un sujeto recibiendo ventilación bajo estos modos. Las flechas indican esfuerzos inspiratorios del paciente, las cuales tan solo introducen perturbaciones al sistema al ser todas las respiraciones mandatorias [26].

En la Fig. 4.2 se pueden ver los resultados de una simulación en el modo VC-CMV, desde la ventana *Monitor* de *VentApp*. La principal diferencia en cuanto a la morfología de estas curvas con las de la Fig. 4.1-A se aprecia entre las fases inspiratoria y espiratoria. Esto es debido a que el ventilador implementado en el caso real produce una pausa inspiratoria, lo cual provoca que la presión en la vía aérea se establezca en un nivel menor que el de pico antes de comenzar la espiración, la denominada presión meseta. Sin embargo, no se considera que esta diferencia sea significativa pues está basada en la implementación de una funcionalidad concreta en el ventilador del caso real, y no en un mal desempeño del simulador. Además, esta funcionalidad no afecta a las curvas a lo largo del resto de la respiración, que se aprecian son muy similares entre los dos casos. Se observa también como, ante un esfuerzo inspiratorio, el flujo espiratorio se ve reducido o incluso detenido, y como este también se ve reflejado en la presión en la vía aérea, evidenciando aún más las similitudes entre ambos resultados. El ruido que presentan las señales de la Fig. 4.1 puede considerarse despreciable, pues este es intrínseco de los entornos reales, no así de los simulados. En cuanto a los niveles de flujo y de presión, no se puede realizar una comparación exhaustiva, puesto que no se dispone de los datos de resistencia y elastancia del paciente real. Sin embargo, puede apreciarse que son del mismo orden de magnitud. Por todo lo expuesto, se considera que el modo VC-CMV se ha implementado de manera suficientemente rigurosa.

En cuanto al modo PC-CMV, se muestran los resultados de la simulación en la Fig. 4.3. Las mismas consideraciones aplican a la hora de compararlos con las curvas de la Fig. 4.1-B en relación al ruido de las señales y el esfuerzo inspiratorio del paciente. En cuanto a la morfología de las curvas, estas presentan formas de onda muy similares. Del mismo modo, la magnitud de las variables son muy semejantes, por lo que se puede concluir también que el modo PC-CMV ha sido implementado correctamente.

### 4.1.2. PAV

Con el objetivo de validar el modo *Proportional Assist Ventilation*, se recurre al ensayo clínico descrito en [22]. Pese a no disponer de datos concretos, se recrea el esfuerzo inspiratorio graficado en la Fig. 4.4 y, al igual que en los casos anteriores, se buscan similitudes entre las formas de onda del caso experimental y del simulado (Fig. 4.5). Además, en la Fig. 4.4 se puede ver también la respuesta de un ventilador en modo PSV ante un esfuerzo inspiratorio similar, de manera que se pueden apreciar más fácilmente las diferencias entre ambos modos.

Comparando ambos resultados, se aprecian formas realmente similares. Como principales diferencias, está el tiempo de respuesta del ventilador desde que detecta el descenso de la presión en la vía aérea, causado por el esfuerzo inspiratorio del paciente, hasta que comienza a generar la presión necesaria para alcanzar la consigna. Además, también se observa un transitorio mayor en el descenso de la presión al comienzo de la fase espiratoria, llegando incluso a *rebotar* en cierto momento. Esto es debido a la forma en que se modela el esfuerzo inspiratorio del paciente, concretamente la fase final. Un tiempo de

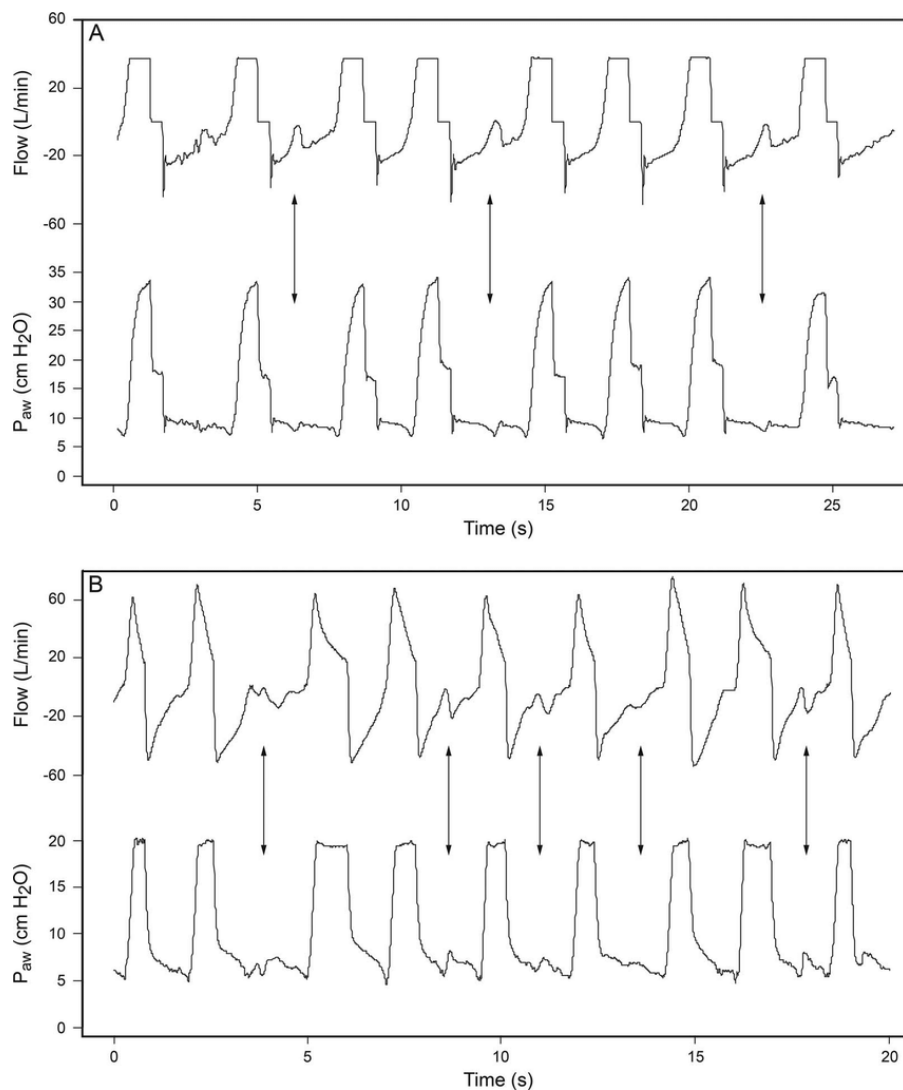


Fig. 4.1: Curvas de flujo y presión en un paciente real recibiendo ventilación en modo VC-CMV (A) y PC-CMV (B) [26]



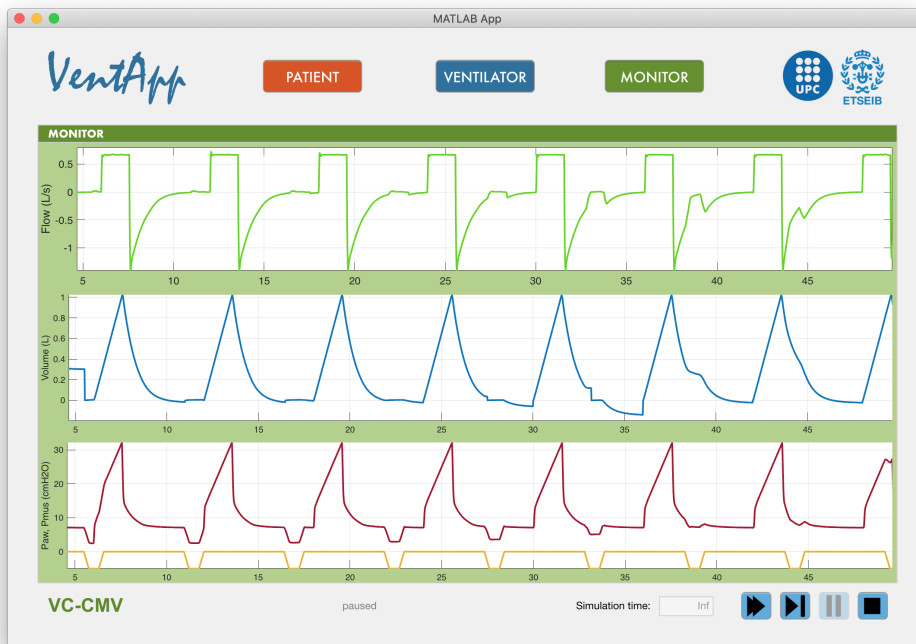


Fig. 4.2: Curvas de flujo, volumen y presión en la vía aérea obtenidas del simulador desarrollado en el modo VC-CMV [elaboración propia]



Fig. 4.3: Curvas de flujo, volumen y presión en la vía aérea obtenidas del simulador desarrollado en el modo PC-CMV [elaboración propia]

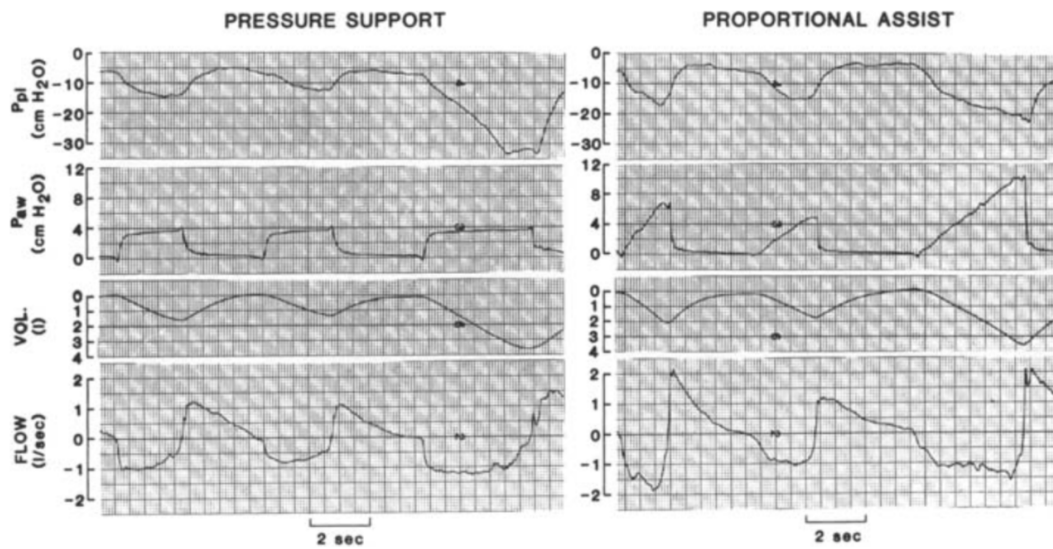


Fig. 4.4: Curvas de esfuerzo inspiratorio ( $P_{pl}$  = pleural pressure), presión en la vía aérea, volumen y flujo en un paciente real recibiendo ventilación en modo PSV (*Pressure Support*) y PAV (*Proportional Assist*) [22]



Fig. 4.5: Curvas de esfuerzo inspiratorio, presión en la vía aérea, volumen y flujo en un paciente real recibiendo ventilación en modo PAV [22]

relajación de los músculos demasiado largo provoca que la presión en boca descienda más lentamente, y que además el valor máximo del flujo espiratorio sea menor. Podemos atribuir estas diferencias, por

tanto, al diseño del controlador y a la modelización del esfuerzo inspiratorio. Pese a esto, se considera que los resultados de la simulación se ajustan con suficiente precisión a los esperados.

## 4.2. VALIDACIÓN CUANTITATIVA

Para una comprobación más rigurosa del funcionamiento del modelo implementado, se ha realizado un procedimiento de validación cuantitativa para el modo Pressure Support Ventilation. Para ello, se ha realizado una prueba de contraste estadístico comparando los resultados fruto de simulaciones del modelo y los obtenidos de procedimientos experimentales. Para estos últimos, se ha contado con una base de datos registrada en el marco de proyectos de investigación, cortesía del Grupo de Investigación en Bioinstrumentación e Ingeniería Clínica (GIBIC), de la Universidad de Antioquia.

### 4.2.1. Datos experimentales

La base de datos utilizada ha sido la de *Sujetos sanos bajo ventilación mecánica* [27]. Se trata de datos experimentales de 41 voluntarios varones sanos ( $26.5 \pm 5.3$  años,  $172.6 \pm 4.0$  cm de altura,  $74.2 \pm 9.6$  kg de peso, índice de masa corporal inferior a  $30$  kg/m<sup>2</sup> y etnia hispana) sometidos a dos pruebas experimentales de ventilación mecánica no invasiva. Sujetos bajo ningún tratamiento médico y sin traumas torácicos. Ninguno tenía marcapasos u otros tipos de estimuladores eléctricos implantados. Fueron excluidos del estudio aquellos sujetos que hubieran consumido alcohol o sustancias alucinógenas 48 horas antes del registro o practicantes de yoga o pilates.

Dos experimentos fueron diseñados para analizar el trabajo dinámico respiratorio en sujetos sanos ante diferentes configuraciones de PEEP y presión soporte (PS) en el modo de ventilación PSV. En ambos experimentos, los sujetos fueron sometidos a ventilación mecánica no invasiva utilizando un ventilador G5 de la marca Hamilton. Ambos experimentos partieron de la siguiente configuración inicial:

- Presión soporte: 0 cmH<sub>2</sub>O
- PEEP: 0 cmH<sub>2</sub>O
- Sensibilidad del disparo inspiratorio (PTTrigger): 0.5 cmH<sub>2</sub>O (configuración por defecto).
- ETS: 25 %
- Fracción inspirada de O<sub>2</sub>: 21 %

Para la validación de este proyecto tan solo se han utilizado los datos del segundo experimento, el cual consistió en variaciones de la PS desde 0 hasta 10 cmH<sub>2</sub>O con incrementos de 2 cmH<sub>2</sub>O cada 3 minutos. En cada incremento, el primer minuto fue considerado la fase de adaptación ante cada cambio de configuración, mientras que los dos minutos restantes fueron usados para el análisis de las señales. En cada registro, maniobras de oclusión de la vía aérea al final de la inspiración fueron configuradas en el ventilador para la estimación a posteriori de la distensibilidad pulmonar (o compliancia) del paciente.

Para cada sujeto fueron registradas, utilizando un datalogger conectado al ventilador mecánico a una frecuencia de muestreo de 1024, las siguientes señales ventilatorias:

- presión en la vía aérea ( $P_{aw}$ ),
- flujo aéreo ( $\dot{V}(t)$ )y
- volumen circulante ( $V(t)$ ).

También fueron registradas variables electromiográficas de superficie (sEMG) de músculos respiratorios (diafragma, esternocleidomastoideo e intercostales), no obstante, estas últimas no son consideradas en este proyecto debido a las limitaciones del mismo. La distensibilidad pulmonar (C) en cada paciente

fue estimada, a posteriori, a partir de los valores de presión y volumen, obtenidos en cada una de las maniobras de oclusión, con la siguiente relación:

$$C = \frac{V}{P} \quad (4.1)$$

Adicionalmente, tanto la resistencia respiratoria ( $R$ ) como la amplitud de la presión muscular ( $P_{mus}$ ) de cada sujeto fueron estimados utilizando el algoritmo de optimización de programación cuadrática secuencial (SQP) y la ecuación de movimiento (2.2) [28].

## 4.2.2. Metodología

Para llevar a cabo este proceso se ha realizado una simulación del modelo configurado en modo PSV para cada muestra de datos disponibles (41 sujetos x 6 niveles de presión soporte = 246 casos). En cada simulación se definen los parámetros de entrada del modelo a partir de la base de datos. Estos comprenden tanto los relacionados con el modo ventilatorio y controles del ventilador como los relativos al esfuerzo muscular y a la mecánica ventilatoria de cada sujeto (ver Tabla 4.1)

	Modo ventilatorio	PSV
Ventilador	Parámetros de control	Presión soporte
		Rampa de presión
		Umbral por presión de disparo inspiratorio
		ETS
		PEEP
Sistema respiratorio	Esfuerzo muscular	Amplitud
		Frecuencia respiratorio
	Mecánica ventilatoria	Resistencia
		Compliancia

Tabla 4.1: Parámetros de entrada al modelo en el proceso de validación [elaboración propia]

Los resultados de las simulaciones son las correspondientes señales de presión en la vía aérea, flujo y volumen circulante. Con el fin de facilitar la comparación entre valores experimentales y simulados, las siguientes características fueron extraídas, ciclo a ciclo, de señales ventilatorias estables (tanto de señales experimentales preseleccionadas como de variables simuladas) para cada sujeto:

- Flujo de pico inspiratorio (L/s)
- Flujo espiratorio medio (L/s)
- Tiempo inspiratorio (s)
- Tiempo espiratorio (s)
- Pausa inspiratoria (s)
- Volumen inspirado (L)
- Volumen espirado (L)
- Pico presión inspiratoria (cmH<sub>2</sub>O)
- Presión espiratoria media (cmH<sub>2</sub>O)

- Presión media (cmH<sub>2</sub>O)
- Presión meseta (plateau) (cmH<sub>2</sub>O)
- Trabajo inspiratorio medio (J/L)
- Trabajo espiratorio medio (J/L)

Para la extracción de características fue utilizada una versión adaptada para este proyecto del Software BreathMetrics, una Toolbox de MATLAB desarrollada por el Laboratorio de Neurociencia Humana de la Universidad Northwestern para el análisis de registros de señales ventilatorias [29]. El contraste estadístico del modelo, basado en estas características, ha sido realizado gracias al *software* desarrollado por la profesora y tutora de este proyecto, Leidy Yanet Serna Higueta [27].

### 4.2.3. Resultados y discusiones

Los resultados obtenidos del proceso de validación se presentan en la Fig. 4.6, donde se muestran diagramas de cajas de las características extraídas tanto de los datos experimentales como de los simulados.

De estos datos se obtiene que tan solo la duración del ciclo respiratorio y la presión (inspiratoria) media no muestran diferencias estadísticas significativas ( $p$ -value  $>0.05$ ). Sin embargo, se observa que la gran mayoría de características presentan un porcentaje de error muy bajo, con tan solo dos características por encima del 20%; e incluso algunas de las más elocuentes, como la presión inspiratoria, con errores realmente pequeños. Teniendo en cuenta también las similitudes entre las distribuciones obtenidas, y la alta variabilidad inherente a las señales fisiológicas, como son estas, se considera que estos datos son considerablemente buenos. Para arrojar algo más de luz en torno a esto y a la presión espiratoria, y por consiguiente al trabajo espiratorio (las características con error considerable), se muestran también las señales reales y simuladas obtenidas de un caso aleatorio (Fig. 4.7).

Se puede apreciar que todas las curvas comparten la misma morfología entre el caso real y el simulado, aunque sí se evidencia un desfase en la presión espiratoria. Esto puede tener relación con el hecho de que el tiempo inspiratorio sea ligeramente menor, pues para la presión muscular dada, existe un retraso desde que el modelo del ventilador cicla la inspiración hasta que el paciente relaja la musculatura. Dado que el esfuerzo inspiratorio introduce una presión negativa en la  $P_{aw}$ , cuando este cesa produce un incremento. Esto, debido al desfase temporal mencionado, se da ya en la fase espiratoria, la cual es pasiva y no dispone de ningún mecanismo de control de la presión más allá de la PEEP, por lo que el efecto es mucho más notorio.

Con todo, se concluye que el modelo es capaz de replicar con suficiente nivel de realismo y precisión experiencias ventilatorias reales.

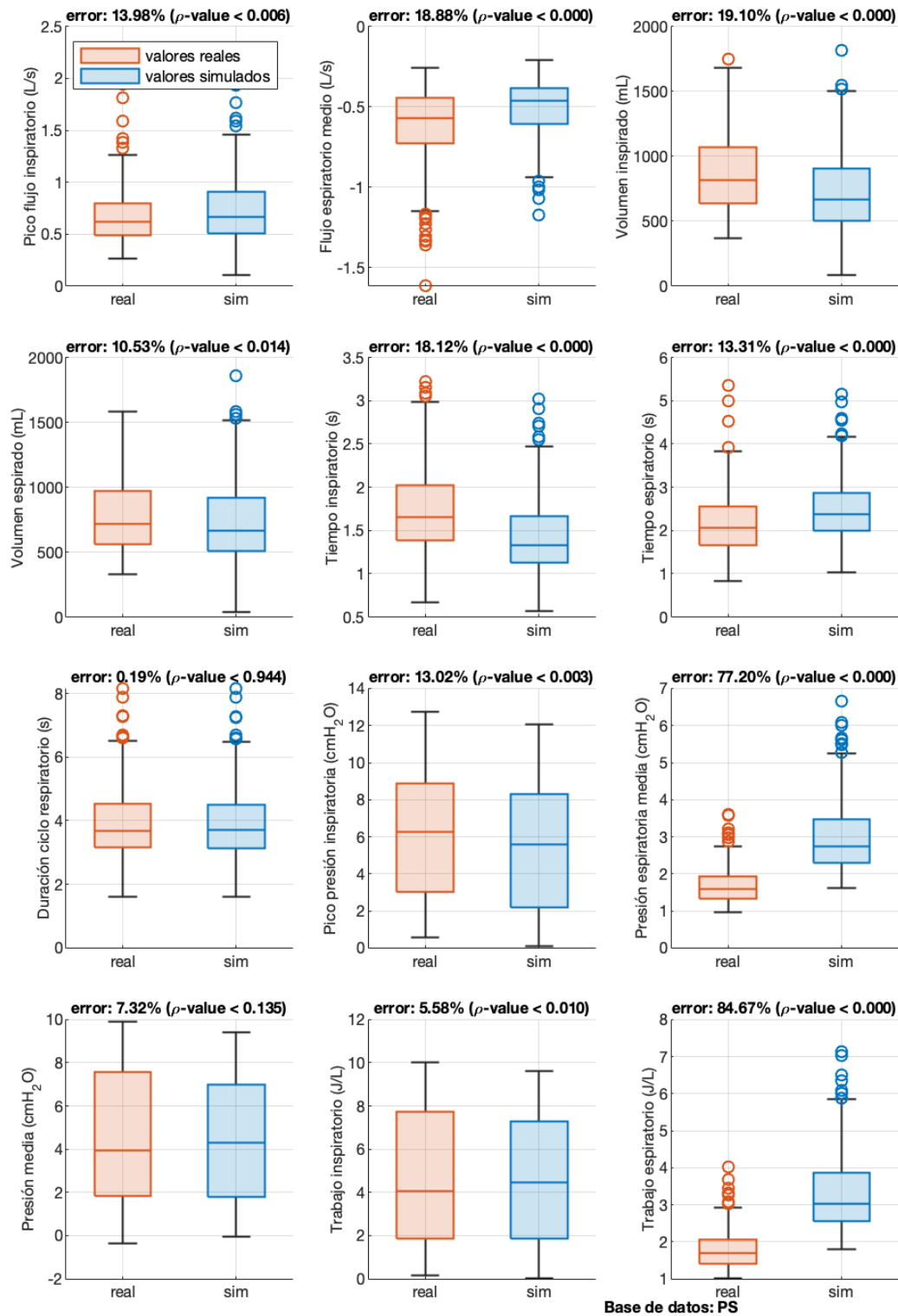


Fig. 4.6: Resultados del proceso de validación del modo PSV [elaboración propia]

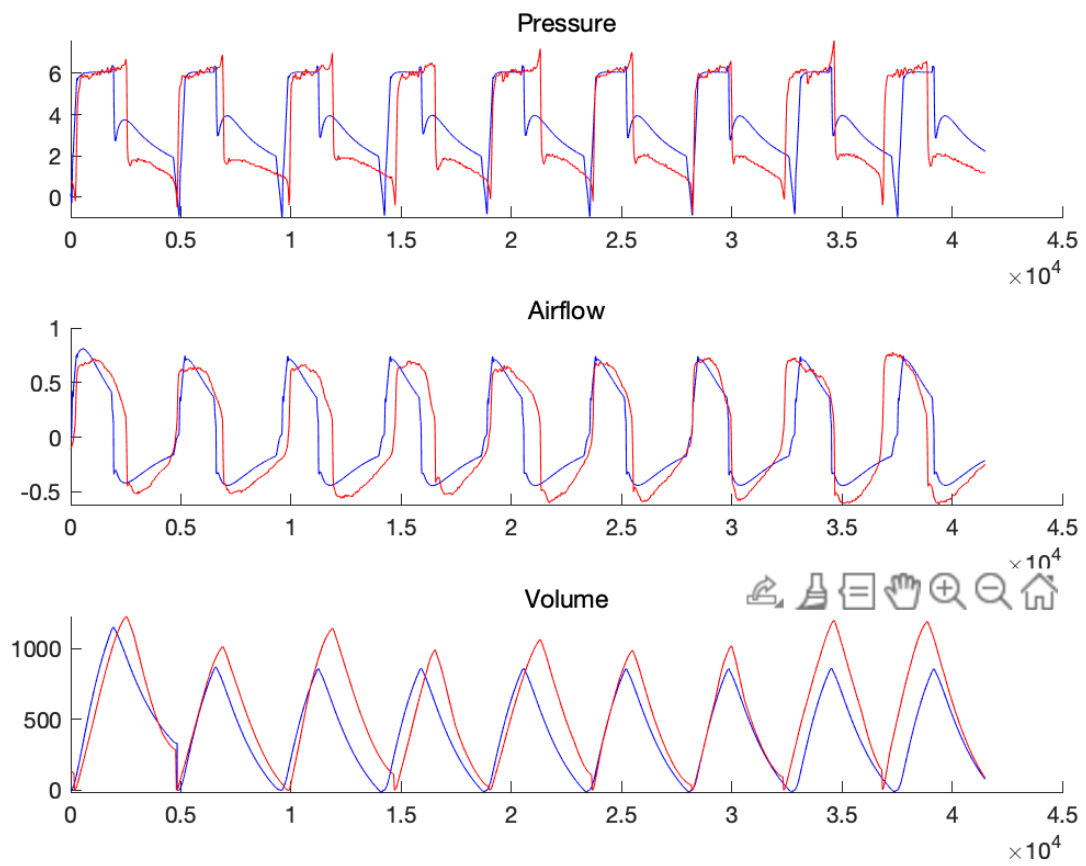


Fig. 4.7: Curvas reales (rojo) y simuladas (azul) de presión en la vía aérea, flujo y volumen de un sujeto aleatorio ( $P_{soporte} = 6 \text{ cmH}_2\text{O}$ ) [elaboración propia]





## Valoración económica

En este capítulo se muestra la valoración económica derivada de las actividades realizadas como parte del presente proyecto. En la Tabla 5.1 se recogen los distintos costes generados en torno a materiales, licencias y mano de obra.

Código	Unidades	Concepto	Medición	Precio	Importe
01		MATERIALES	1	1.500,00 €	1.500,00 €
01.01	Ud.	MacBook Pro Retina 13"	1	1.500,00 €	1.500,00 €
02		LICENCIAS	1	89,00 €	89,00 €
02.01	Ud.	Licencia MATLAB and Simulink Student Suite	1	69,00 €	69,00 €
02.02	Ud.	Licencia Simscape	1	20,00 €	20,00 €
03		MANO DE OBRA	1	5.040,00 €	5.040,00 €
03.01	h	Estudio de los fundamentos de la ventilación mecánica	45	14,00 €	630,00 €
03.02	h	Estudio del modelo de partida y las herramientas utilizadas	18	14,00 €	252,00 €
03.03	h	Desarrollo y validación del modelo	102	14,00 €	1.428,00 €
03.04	h	Desarrollo y validación de la interfaz gráfica	80	14,00 €	1.120,00 €
03.05	h	Redacción de la memoria	115	14,00 €	1.610,00 €
<b>COSTE DEL PROYECTO SIN IVA</b>					<b>6.629,00 €</b>
IVA (21%)					1.392,09 €
<b>COSTE DEL PROYECTO CON IVA</b>					<b>8.021,09 €</b>

Tabla 5.1: Valoración económica del proyecto [elaboración propia]

Asciende el coste total del proyecto a la cantidad de OCHO MIL VEINTIÚN EUROS CON NUEVE CÉNTIMOS, IVA incluido.



# Impacto ambiental

En este capítulo se estiman las afecciones en materia de impacto ambiental derivadas de la realización de este proyecto.

Las consecuencias medioambientales de este proyecto se derivan principalmente del consumo, tanto de la energía eléctrica necesaria para llevar a cabo el proyecto como del material utilizado para ello.

Así pues, diversos estudios apuntan a que la fabricación de un ordenador portátil genera alrededor de unos 250 kilogramos de CO<sub>2</sub>. En cuanto al consumo, la cantidad se establece en unos 200 gramos de CO<sub>2</sub> a la hora, lo cual resulta en un total de 72 kilogramos de CO<sub>2</sub> en la realización de este trabajo, ascendiendo el total de la huella de carbono a 322 kilogramos de CO<sub>2</sub>.

En compensación, la solución desarrollada permite ahorrar un considerable, aunque difícil de estimar, consumo de energía y de recursos al sustituir en ciertas ocasiones el uso de ventiladores mecánicos reales por el del simulador realizado en este proyecto.



# Conclusiones

## 7.1. CUMPLIMIENTO DE OBJETIVOS

El objetivo de este trabajo era el de proporcionar un simulador de ventilación mecánica basado en un modelo de ventilador con suficiente nivel de detalle para ser objeto de estudio y mejora por parte de ingenieros y técnicos, implementando las funcionalidades básicas de un simulador de estas características para hacer del conjunto de la solución una experiencia lo más realista posible.

Se ha tomado como punto de partida un modelo de ventilador mecánico descrito matemáticamente a nivel de cada elemento que integra un ventilador real de presión positiva. Sobre este modelo se han implementado cuatro modos ventilatorios, a cada cual más novedoso y complejo, que además facilitan la introducción del usuario del simulador a los fundamentos de la ventilación. Así mismo, se ha desarrollado una interfaz gráfica que persigue simular una experiencia ventilatoria real, permitiendo a su vez interactuar con el modelo de forma más intuitiva y sencilla.

El correcto funcionamiento del modelo ha sido verificado en sus cuatro modos ventilatorios, habiendo contado con datos experimentales para una validación más exhaustiva en el caso del modo *Pressure Support Ventilation*. Igualmente, se ha comprobado el buen funcionamiento de la interfaz gráfica y su interacción con el modelo.

En resumen, el resultado final de este trabajo es un modelo detallado de un ventilador mecánico que implementa con rigor cuatro modos ventilatorios: dos de respiraciones mandatorias, con control de volumen (VC-CMV) y de presión (PC-CMV), respectivamente; uno de control de presión con respiraciones espontáneas (PSV); y otro que proporciona una asistencia proporcional al esfuerzo instantáneo del paciente (PAV). El modelo que integra estos cuatro modos ventilatorios es completamente accesible y modificable. Además, es escalable, de forma que permite la adición de nuevos modos o incluso diseños del ventilador conservando la estructura modular de la solución, facilitando así, además, la integración de estas nuevas versiones en la interfaz gráfica. Esta interfaz también es parte del resultado final del trabajo. Una vez probados e implementados los diseños del ventilador, la aplicación permite experimentar con estos de forma cómoda y realista, como si se tratase de un ventilador real.

Se concluye, por tanto, que los objetivos del trabajo han sido conseguidos con un alto grado de éxito y satisfacción.

## 7.2. FUTURO DEL PROYECTO

El trabajo realizado abre un gran número de puertas a la investigación del funcionamiento y al diseño de los ventiladores mecánicos. Los trabajos futuros que surgen de este proyecto abarcan desde la optimización en el diseño de los componentes del ventilador, hasta la experimentación con nuevos e innovadores modos de ventilación.

En cuanto a las limitaciones de la solución desarrollada, también hay una serie de trabajos que podrían desarrollarse. Principalmente, en relación al diseño del controlador. Hoy en día, los ventiladores mecánicos más modernos incorporan las técnicas más punteras en controladores de tipo adaptativo, que cambian sus parámetros constantemente para ajustarse al entorno cambiante. Un controlador de este tipo podría corregir las diferencias observadas en el proceso de validación, o quizás un controlador

con otros parámetros permita acercarse a una dinámica deseada concreta. Del mismo modo, existe un amplio abanico surgiendo en torno a la inteligencia artificial y los ventiladores mecánicos, que no son ajenos a estas nuevas tecnologías.

En definitiva, tal y como la situación extraordinaria de pandemia bajo la cual se desarrolla este proyecto ha hecho notar, desgraciadamente, la ventilación mecánica es una técnica de vital importancia en el entorno hospitalario que, lejos de haber alcanzado su culmen, tiene aún muchos avances que ofrecer.

# Glosario

**CMV** Secuencia respiratoria en la que no se permiten respiraciones espontáneas al ser todas mandatorias. 16, 21

**Compliancia pulmonar ( $C$ )** Capacidad de los pulmones de aumentar de volumen ante un aumento de la presión. Es un anglicismo, también se denomina distensibilidad y es la magnitud inversa de la elastancia. 5, 6, 11, 21

**Constante de tiempo ( $\tau$ )** Expresión matemática que describe la eficacia relativa del llenado y vaciado de los pulmones, calculado como la división de la resistencia por la elastancia y medida en segundos. 6, 7

**CPAP** Modo de ventilación en el que se proporciona una presión positiva continua en las vías respiratorias (*Continuous Positive Airway Pressure*). 16

**CSV** Secuencia respiratoria en la que todas las respiraciones son espontáneas. 16, 21

**Elastancia pulmonar ( $E$ )** Capacidad de los pulmones de recuperar sus dimensiones iniciales ante un cese de la deformación por presión. Magnitud inversa a la compliancia o distensibilidad y comparable a la rigidez. 6, 23–25, 30, 35

**ETS** *Expiratory Trigger Sensitivity*. Porcentaje del valor máximo del flujo inspiratorio que, en el modo ventilatorio de presión soporte, marca el comienzo de la fase espiratoria. 15, 16, 22

**Frecuencia respiratoria (FR)** Número de respiraciones realizadas por minuto. 17, 18, 21

**IMV** Secuencia respiratoria en la que se pueden intercalar respiraciones espontáneas entre las mandatorias. 16, 18

**Insuficiencia respiratoria (IR)** Afección en la cual el nivel de oxígeno en sangre es demasiado bajo o el de dióxido de carbono demasiado alto, o ambos. 4

**PAV** *Proportional Assist Ventilation*. Modo de ventilación controlado por presión, en la que todas las respiraciones son espontáneas y que sigue un esquema de control del tipo servo, ajustando su salida al esfuerzo inspiratorio instantáneo del paciente. 22, 34

**PC-CMV** *Pressure Controlled - Continuous Mandatory Ventilation*. Modo de ventilación controlado por presión y en la que todas las respiraciones son mandatorias y cicladas por tiempo. 21, 33

**PEEP** *Positive End-Expiratory Pressure*. Presión positiva al final de la espiración, referida normalmente a la PEEP extrínseca. 7, 15, 17, 21, 26

**PEEP extrínseca** Presión en la vía aérea al final de la espiración, aplicada por el ventilador. 7, 15, 26

**PEEP intrínseca** Diferencia de presión positiva entre los alveolos y la vía aérea al final de la espiración, causada por una exhalación incompleta. También denominada auto-PEEP. 7, 15, 26

**Presión de control ( $P_{control}$ )** Parámetro configurable en los modos de control de presión que establece el valor máximo de presión en la vía aérea durante la inspiración, también denominado presión soporte. 21

**Presión en la vía aérea ( $P_{aw}$ )** Presión medida en boca, al comienzo de la vía aérea. 26

**Presión muscular ( $P_{mus}$ )** Presión generada por los músculos respiratorios. 5, 6, 12, 23–25, 30

**Presión transrespiratoria ( $P_{TR}$ )** Gradiente de presión entre la vía aérea superior y la superficie corporal, que determina la cantidad de flujo inspiratorio. 5, 6

- Presión ventilada ( $P_{vent}$ )** Presión generada por el ventilador mecánico. 6, 21–25
- PSV** *Pressure Support Ventilation*. Modo de ventilación controlado por presión y en la que todas las respiraciones son espontáneas. 15, 16, 18, 21, 33
- Resistencia de la vía aérea ( $R$ )** Resistencia que ofrece el árbol tranqueobronquial al paso del aire a través de él. 5, 6, 21, 23, 25, 30, 35
- Respiración asistida** Respiración en la que el ventilador mecánico realiza algún tipo de trabajo para el paciente. 16
- Respiración espontánea** Respiración disparada y ciclada por el paciente. 16
- Respiración mandatoria** Respiración en la que el paciente no tiene control ni del tiempo ni de las dimensiones. El ventilador marca el comienzo y/o fin de la inspiración. 16
- Tiempo de subida ( $P_{ramp}$ )** Parámetro configurable en los modos de control de presión que establece el tiempo que tarda el ventilador en alcanzar la presión de control de forma lineal desde el nivel de presión de base, una vez comienza la inspiración. 21
- Tiempo espiratorio ( $T_e$ )** Duración de la fase espiratoria durante una respiración. 18
- Tiempo inspiratorio ( $T_i$ )** Duración de la fase inspiratoria durante una respiración. 10, 15, 17, 18, 21, 23
- VC-CMV** *Volume Controlled - Continuous Mandatory Ventilation*. Modo de ventilación controlado por volumen y en la que todas las respiraciones son mandatorias y cicladas por tiempo. 21, 33, 34
- Ventilador mecánico (VM)** Dispositivo encargado de dar soporte de manera artificial a la ventilación de un paciente con insuficiencia respiratoria. 3–14, 16–18, 33, 34
- Ventilación mecánica invasiva (VMI)** Modalidad ventilatoria en la que se aplica una presión positiva a través de un tubo endotraqueal o de una traqueostomía. 10, 11
- Ventilación mecánica no invasiva (VMNI)** Modalidad ventilatoria en la que se aplica una presión positiva a través de una interfaz colocada encima de la boca, la nariz o el rostro entero, sin invadir la vía respiratoria. 10, 11
- Volumen minuto ( $V_M$ )** Volumen de gas desplazado en la respiración durante un minuto. Función del volumen tidal y de la frecuencia respiratoria. También llamado ventilación minuto. 16, 18
- Volumen tidal ( $V_T$ )** Volumen de gas desplazado durante un ciclo respiratorio. También llamado volumen corriente. 11, 14, 17, 21, 23



# Bibliografía

- [1] G. L. Colice, “Historical Perspective on the Development of Mechanical Ventilation,” en *Principles and Practice of Mechanical Ventilation*, M. J. Tobin, Ed., 3.<sup>a</sup> ed., The McGraw-Hill Companies, Inc., 2013.
- [2] A. S. Slutsky, “History of Mechanical Ventilation. From Vesalius to Ventilator-induced Lung Injury,” *American Journal of Respiratory and Critical Care Medicine*, vol. 191, págs. 1106-1115, 2015. doi: 10.1164/rccm.201503-0421pp.
- [3] R. M. Kacmarek, “The Mechanical Ventilator: Past, Present, and Future,” *Respiratory Care*, vol. 56, n.º 8, págs. 1170-1180, ago. de 2011. doi: 10.4187/respcare.01420.
- [4] Polio Survivors Association. “Pictures of Polio.” (2021), [En línea]. Disponible en: <http://www.polioassociation.org/pictures.html> [Accedido: 14-07-2021].
- [5] Maquet Critical Care AB. “Catálogo Ventilador Mecánico Servo-u.” (2019), [En línea]. Disponible en: [shorturl.at/apLOT](http://shorturl.at/apLOT) [Accedido: 14-07-2021].
- [6] L. A. Ramos Gómez y S. Benito Vales, *Fundamentos de la ventilación mecánica*, 1.<sup>a</sup> ed. Barcelona: Marge Médica Books, 2012, cap. 4, págs. 53-55.
- [7] MedlinePlus en español. “Insuficiencia respiratoria,” Bethesda (MD): Biblioteca Nacional de Medicina (EE. UU.) (2020), [En línea]. Disponible en: <https://medlineplus.gov/spanish/respiratoryfailure.html> [Accedido: 15-07-2021].
- [8] F. Gutiérrez Muñoz, “Ventilación mecánica,” *Acta Médica Peruana*, vol. 28, págs. 87-104, abr. de 2011, ISSN: 1728-5917. [En línea]. Disponible en: [http://www.scielo.org.pe/scielo.php?script=sci\\_arttext&pid=S1728-59172011000200006&nrm=iso](http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1728-59172011000200006&nrm=iso).
- [9] R. L. Chatburn, “Classification of Mechanical Ventilators and Modes of Ventilation,” en *Principles and Practice of Mechanical Ventilation*, M. J. Tobin, Ed., 3.<sup>a</sup> ed., The McGraw-Hill Companies, Inc., 2013.
- [10] T. Similowski y J. Bates, “Two-compartment modelling of respiratory system mechanics at low frequencies: gas redistribution or tissue rheology?” *European Respiratory Journal*, vol. 4, n.º 3, págs. 353-358, 1991, ISSN: 0903-1936. [En línea]. Disponible en: <https://erj.ersjournals.com/content/4/3/353>.
- [11] L. Y. Serna Higuera, *Respiratory rehabilitation equipment, mechanical ventilation*, Universitat Politècnica de Catalunya, Centre de Recerca de en Enginyeria Biomèdica, 2021.
- [12] R. L. Chatburn y T. A. Volsko. “Mechanical Ventilators.” (2016), [En línea]. Disponible en: <https://thoracickey.com/mechanical-ventilators> [Accedido: 18-07-2021].
- [13] O. Díaz y G. Buggedo, 2016. [En línea]. Disponible en: <http://www.terapiaventilatoria.uc.cl/Articulos/ventilacion-mecanica-en-el-paciente-con-obstruccion-bronquial>.
- [14] R. L. Chatburn y E. Mireles-Cabodevila, “Basic Principles of Ventilator Design,” en *Principles and Practice of Mechanical Ventilation*, M. J. Tobin, Ed., 3.<sup>a</sup> ed., The McGraw-Hill Companies, Inc., 2013.
- [15] Google, LLC. “Google imágenes.” (2021), [En línea]. Disponible en: [google%20imagenes%20no%20funciona](https://www.google.com/search?q=google%20imagenes%20no%20funciona) [Accedido: 23-07-2021].
- [16] R. Adasme y P. Figueroa, “Recomendaciones de armado y uso de filtros de Ventilación Mecánica COVID19,” abr. de 2020.
- [17] A. Orlando, “How to improve patient-ventilator synchrony. Waveform analysis and optimization of ventilator settings,” Hamilton Medical AG, 2018.
- [18] M. B. P. Amato y J. J. Marini, “Pressure-Controlled and Inverse-Ratio Ventilation,” en *Principles and Practice of Mechanical Ventilation*, M. J. Tobin, Ed., 3.<sup>a</sup> ed., The McGraw-Hill Companies, Inc., 2013.
- [19] MedlinePlus en español. “Tratamiento de presión positiva en las vías respiratorias,” Bethesda (MD): Biblioteca Nacional de Medicina (EE. UU.) (2021), [En línea]. Disponible en: <https://medlineplus.gov/spanish/ency/article/001916.htm> [Accedido: 30-07-2021].

- [20] A. J. Garnero, H. Abbona, F. Gordo-Vidal y C. Hermosa-Gelbard, “Modos controlados por presión versus volumen en la ventilación mecánica invasiva,” *Medicina Intensiva*, vol. 37, n.º 4, págs. 292-298, 2013. doi: 10.1016/j.medin.2012.10.007.
- [21] M. Younes, “Proportional Assist Ventilation, a New Approach to Ventilatory Support: Theory,” *American Review of Respiratory Disease*, vol. 145, n.º 1, págs. 114-120, 1992, PMID: 1731573. doi: 10.1164/ajrccm/145.1.114.
- [22] M. Younes, A. Puddy, D. Roberts y col., “Proportional Assist Ventilation: Results of an Initial Clinical Trial,” *American Review of Respiratory Disease*, vol. 145, n.º 1, págs. 121-129, 1992, PMID: 1731574. doi: 10.1164/ajrccm/145.1.121.
- [23] The MathWorks, Inc. “Productos y Servicios.” (2021), [En línea]. Disponible en: [https://es.mathworks.com/products.html?s\\_tid=gn\\_ps](https://es.mathworks.com/products.html?s_tid=gn_ps) [Accedido: 08-08-2021].
- [24] —, “Medical Ventilator with Lung Model.” (2021), [En línea]. Disponible en: <https://es.mathworks.com/help/physmod/simscape/ug/medical-ventilator-with-lung-model.html> [Accedido: 08-08-2021].
- [25] S. Roy, L. Bunting, S. Stahl y D. Textor, “Inline Positive End-Expiratory Pressure Valves: The Essential Component of Individualized Split Ventilator Circuits,” *Critical Care Explorations*, vol. 2, e0198, sep. de 2020. doi: 10.1097/cce.000000000000198.
- [26] C. Subira, C. de Haro, R. Magrans, R. Fernandez y L. Blanch, “Minimizing Asynchronies in Mechanical Ventilation: Current and Future Trends,” *Respiratory Care*, vol. 63, respcare.05949, feb. de 2018. doi: 10.4187/respcare.05949.
- [27] L. Y. Serna Higueta, “Validación de contraste estadístico,” Universitat Politècnica de Catalunya, 2020.
- [28] I. C. Muñoz y A. M. Hernández, “Noninvasive approach to estimate ventilatory mechanics in spontaneous breathing with different PEEP and pressure support values: validation with mechanical simulation,” en *VII Latin American Congress on Biomedical Engineering CLAIB 2016, Bucaramanga, Santander, Colombia, October 26th -28th, 2016*, I. Torres, J. Bustamante y D. A. Sierra, Eds., Singapore: Springer Singapore, 2017, págs. 241-244.
- [29] T. Noto, G. Zhou, S. Schuele, J. Templer y C. Zelano, “Automated analysis of breathing waveforms using BreathMetrics: a respiratory signal processing toolbox,” *Chemical Senses*, vol. 43, n.º 8, págs. 583-597, jul. de 2018, ISSN: 0379-864X. doi: 10.1093/chemse/bjy045. eprint: <https://academic.oup.com/chemse/article-pdf/43/8/583/25822269/bjy045.pdf>. [En línea]. Disponible en: <https://doi.org/10.1093/chemse/bjy045>.