POLITECNICO DI MILANO

Scuola di Ingegneria dei Sistemi

Corso di Laurea in

Ingegneria Biomedica

A NEW SYSTEM TO MONITOR CHANGES OF LUNG MECHANICS DURING RESUSCITATION OF NEWBORNS BY FOT

Relatore: Prof. Dellacà Raffaele

Correlatore: Ing. Veneroni Chiara

Ing. Zannin Emanuela

Tesi di Laurea di:

Alberto MOTTA matr. 766020

Anno Accademico 2011-2012

A Marco ed Alice.

Che questo mio grande traguardo,

sia solo l’inizio dei Nostri.

Table of Contents

SUMMARY

At birth, lung ventilation becomes the first step for respiration and it is fundamental to provide the correct amount of oxygen to the tissues.

Birth represents a critical event associated with dramatic changes in the lung function: the fetal lung fluid which maintains the lungs in a distended state during fetal life needs to be absorbed, the lungs have to be filled with air, adequate gas-exchanging surface area has to be established, the pulmonary blood flow has to greatly increase and the surfactant system has to ensure the lungs remain expanded by decreasing the alveolar surface tension.

For the onset of ventilation the respiratory muscles have to provide an adequate force to overcome the elastic recoil of lung and chest wall, the viscous friction of air flowing and the forces caused by the visco-elastic behaviour of the lung tissues and the inertial forces.

The infant respiratory system has several physiological and anatomical differences with respect to the adults' one. These differences result to bring more difficulties to overcome a respiratory stress.

Moreover, preterm infants are compounded with an incomplete lung structure.

When, for any reason, the respiratory muscles cannot provide the necessary pressure for respiratory flow, the lung ventilation must be guaranteed externally, by artificial ventilators. Although mechanical ventilation is indispensable for the survival of little patients, excessive tidal volumes and inadequate lung recruitment may contribute to mortality by causing ventilator-induced lung injury – such as VILI.

Nowadays, recruiting techniques are investigated in order to assist directly newborn at birth helping them to remove the fetal fluid from lung and establish an adequate volume recruitment, as recruitment of alveoli could bring to avoid ventilation and thus the eventually derived injury, or at least to reduce the ventilator induced lung injury. A lot of techniques has been proposed to recruit the lung. Between them, SLI –Sustained Lung Inflation, a technique which insufflates air at high pressure peaks for few seconds - is gaining popularity.

However as an excessive stress applied to the lung tissue at birth could damage the lung, monitoring respiratory mechanic during this manoeuvre, could help to define an optimal

approach to maximize the final FRC and to minimize the stress applied to pulmonary tissue.

Forced oscillation technique – FOT – is a non-invasive, versatile method to assess respiratory mechanics even without patient’s cooperation. Therefore it is very useful to study infants who are not capable to direct collaboration. FOT has been proved to be sensitive to alveolar recruitment in an animal model and applicable to the study of lung mechanics in ventilated preterm infants. During forced oscillations, a small amplitude sinusoidal pressure stimulus is applied to the airway opening. The mechanical response of the respiratory system is studied thanks to the total respiratory input impedance (Z_{IN}). Z_{IN}, or Z_{RS}, is a complex number which can be expressed as real part, called resistance (R_{RS}), and imaginary part, called reactance (X_{RS}). Particularly, X_{RS} measured at 5Hz is very sensible to changes in the mechanics of lung periphery and provides accurate information about lung volume recruitment and de-recruitment. A FOT setup usually includes a loud speaker which imposes a sinusoidal wave above the ventilator waveform but FOT can be easily implemented in ventilators. However up to now there are no commercial ventilators that implement this technique and the traditional FOT set up is not suitable for application in a critical situation as during SLI at birth.

The *purpose* of this Thesis work is to develop and validate a FOT setup suitable to evaluate lung changes during SLI resuscitations of newborns. The developed setup exploits a newborn ventilator produced by Acutronic: Fabian HFO. Fabian High Frequency Oscillations module of ventilation has been used as generator of oscillations, as FOT requires. In collaboration with Acutronic, few changes have been implemented in the ventilators’ firmware including a communication protocol to send flow and pressure data through serial port and small changes in data elaboration. In order to avoid excessive ventilators’ firmware changes, we acquired the flow and pressure data from the ventilator with an android tablet and developed an application to permit the clinician to have online computation of R and X. The tablet is a very common device, with low cost, which has an open source program language and guarantees high portability.

The developed android application calculates the respiratory impedance Z_{RS} on line, divided in its real and imaginary components. Thus the two recorded measures – flow and pressure – and the two others derived measures, are stored in a file in order to be observed

also later and sub-sampled to be plotted on line. This setup has the advantages of being user friendly AND compact, as it involves only a tablet and a ventilator. Therefore it can be used in emergency cases, such as in NICU – newborns intensive care units – or during very critical moments such as birth.
In vitro tests were performed to validate the developed setup. After that, the set up was tested *in vivo*. Respiratory impedance was measured in three newborns in the neonatal intensive care unit of Mangiagalli e Regina Elena Hospital in Milan. Infants were studied at baseline with the developed set up and with an already validated, but cumbersom, setup. Using the developed setup lung mechanics were studied during the application of SLI.The aim of this study was to evaluate the feasibility of monitoring changes in respiratory mechanics induced by SLI in preterm newborns with the developed set up.

The above tracted topics are mapped in the 5 major chapters of my Thesis.

The first chapter deals with the fundamental of the respiratory system and with the differences between adults and newborns' one. FOT is presented as a non invasive method to monitor respiratory mechanics.
After these physiological hints, the SLI recruitment manoeuvre and the functioning principles of mechanical ventilator are explained.

In the second chapter instead the hardware and basis of android language programming are described. The Fabian HFOV ventilator has been studied and different ways of implementing FOT have been considered. To minimally change the ventilator firmware, we decided to use the HFO module to generate FOT pressure stimulus as it permits to have a mean average pressure, with oscillations superimposed.
A tablet has been chosen for data elaboration. An android tablet seems to be a good choice because of its low cost, its high portability and because of the possibility to quite easily create a suited application – thanks to open source codes. Hints of object oriented programming language are presented, because android programming language needs these theoretical bases.

In the third chapter, the developed setup is presented. Acutronic Fabian firmware has been modified in order to deliver the flow and the pressure data through serial communication, in a precise bounded data packet.
Moreover the original flow data elaboration procedure in the ventilator was modified to permit the observation of small flow changes around zero. As a time shift between flow and pressure data has been found, a correction was computed to correct for it.
Secondly, this chapter deals with the description of the android application. It records data delivered by the ventilator, calculates the components of respiratory impedance - after compensate for the time-shift between flow and pressure, plots and saves them.

Chapter number four deals with the in vitro tests. Bottles and glass capillaries have been mixed to constitute three different lung models, with different values of Resistance and Reactance and used to test the developed set up.
As to measure Z_{RS}, is necessary to have perfect data synchronization. The correction found in the previous chapter is now tested on different lung model to verify that the delay between flow and pressure is not dependent on the mechanical properties of the used test lung. The validation consists in the comparison of values obtained with the developed setup and with another setup, considered as gold standard. Error between these two measurement in the all the test lungs is lower than 10%, compliant with European FOT guidelines:
Resistance Error % 4.37 ± 3.99 cmH_2O*s/l, Reactance Error % 5.42 ± 4.21 cmH_2O*s/l.
Other tests were performed. It has been tested the possibility of applying FOT during SLI, whether lost of data can occur and the accuracy of Z online computation.

The fifth Chapter deals with in vivo measurements. To test whether the developed system is completely reliable, it was tested even in critical and no ideal conditions. Thus three newborns at Clinica Mangiagalli di Milano, were studied. They have been chosen by clinicians; mean gestational age 31.33 ± 2.3 weeks, mean age after birth: 18 ± 10.3 days and mean weight at birth: 1195 ± 169 g – all females.
The comparison between the values obtained with the "Ventilator and Tablet" setup and a gold standard one was performed. The results permit to proceed in the application of SLI manoeuvre.

These preliminary results highlight the difference of lung mechanics response to SLI maneuver between infants. This suggest that monitoring lung mechanics could help in discerning between lungs that improves or not to SLI and adapt pressure and time of the SLI maneuver according to infants' needs.

In conclusion:

- the set-up allows the measurement of respiratory system impedance during SLI in newborns;
- the set-up is easy to manage by the clinicians during the manoeuvre;
- the set-up and procedure is well tolerated by infants;
- the set-up is suitable to be used in the clinical environment.

The criticality that emerged from these first attempts to monitor lung impedance changes during SLI is the interface with the infant particularly in terms of leaks, which should be avoided as much as possible to get reliable measurements.

Further developments would deal with the generation of oscillations by the control of the PEEP valve, and not using HFOV module.

Another development, could occur on Android application. It could automatically chose stable tracts during the inflation and give a feedback of only reliable values.

Further studies can address the choice of the stimulus frequency. Low frequencies are more sensitive to lung peripheral but the SNR – Signal to Noise Ratio - is low because of interference with the frequency components of the breathing signal. In adults 5Hz represents the best compromised between sensitivity to lung periphery and SNR, however as infants have higher breathing frequencies and higher resonant frequency, an higher stimulus frequency could be a best choice.

SOMMARIO

Alla nascita, la ventilazione polmonare è il primo passo per la respirazione ed è fondamentale che sia fornito il giusto quantitativo di ossigeno ai tessuti. La nascita rappresenta un evento critico associato a drammatici cambiamenti nella funzionalità polmonare: il liquido polmonare fetale che mantiene i polmoni in uno stato disteso durante la gestazione deve essere assorbito, i polmoni devono essere ventilati con aria, deve essere mantenuto un adeguato scambio di gas rispetto alla superficie respiratoria. Il flusso sanguigno polmonare aumenta notevolmente ed il tensioattivo deve garantire che i polmoni rimangano pervi, diminuendo la tensione superficiale alveolare.

I muscoli respiratori per permettere la ventilazione, devono fornire una forza sufficiente a superare il ritorno elastico di polmone e parete toracica, l'attrito viscoso dell'aria che fluisce e le forze di opposizione, causate dal comportamento viscoelastico dei tessuti polmonari e dall'inerzia.

La conoscenza completa dello sviluppo polmonare nei bambini e la comprensione delle peculiarità del polmone del neonato, sono i prerequisiti per la scelta di trattamento corretto per i disturbi respiratori infantili.

Il sistema respiratorio del bambino ha grandi differenze fisiologiche e anatomiche rispetto a quello adulto. Queste differenze sembrano aumentare le difficoltà nel superare uno stress respiratorio. Inoltre, la condizione dei neonati pretermine è nuovamente peggiorata, a causa dell'incompleta struttura polmonare.

Quando, per qualsiasi motivo, i muscoli respiratori non possono fornire la pressione necessaria per il normale flusso respiratorio, la ventilazione polmonare deve essere garantita esternamente, da ventilatori artificiali. Nonostante la ventilazione meccanica sia indispensabile per la sopravvivenza dei piccoli pazienti, eccessivi volumi tidali ed eccessive pressioni sottoposte a livello polmonare possono contribuire alla mortalità provocando un danno polmonare indotto da ventilatore, altrimenti detto VILI.

Oggi, tecniche di reclutamento sono molto studiate per assistere il neonato direttamente alla nascita, aiutandolo a rimuovere il liquido fetale dal polmone e stabilendo un reclutamento adeguato. Il reclutamento alveolare potrebbe portare ad evitare la

ventilazione artificiale e quindi il danno polmonare stesso. Molte tecniche sono state proposte per reclutare il polmone. Tra esse, la SLI (Sustained Lung Inflation o sostenuta inflazione polmonare) una tecnica che fornisce alta pressione per qualche secondo, sta guadagnando popolarità.
SLI si è dimostrata essere una tecnica efficace di reclutamento, tuttavia è fondamentale la conoscenza della meccanica respiratoria durante questa manovra, per definire un approccio ottimale per massimizzare la capacità funzionale residua (CFR) finale e minimizzare la sollecitazione applicata al tessuto polmonare.
La tecnica delle oscillazioni forzate (FOT) è una metodologia non invasiva e versatile per la valutazione della meccanica respiratoria, anche senza la collaborazione del paziente. Per questo è molto utile per studiare i bambini che, per natura, non sono in grado di offrire collaborazione. FOT è stato dimostrato essere sensibile al reclutamento alveolare in un modello animale e applicabile allo studio della meccanica polmonare in neonati pretermine ventilati. Durante le oscillazioni forzate, un piccolo stimolo in pressione, di ampiezza sinusoidale viene applicata l'apertura delle vie aeree. La risposta meccanica del sistema respiratorio è studiata grazie alla impedenza totale di ingresso respiratoria (Z_{IN}). $Z_{IN,}$ o Z_{RS} da impedenza respiratoria, è un numero complesso che può essere espresso come parte reale, chiamata resistenza respiratoria (R_{RS}), e parte immaginaria, detta reattanza (X_{RS}). In particolare, X_{RS} misurata a 5Hz è molto sensibile ai cambiamenti della meccanica della periferia polmonare e fornisce informazioni precise sul reclutamento del volume polmonare e de-reclutamento. Una configurazione FOT solitamente comprende un altoparlante che impone un'onda sinusoidale sopra la forma d'onda del ventilatore. Nonostante le oscillazioni forzate possano essere facilmente implementate in un ventilatore, tuttavia fino ad oggi non ci sono ventilatori commerciali con questa tecnica ed il set-up tradizionale (quello con implementazione esterna) non è certamente adatto per uno studio in una situazione critica come durante l’applicazione della SLI alla nascita.

Lo *scopo* di questo elaborato di tesi è, come indica il titolo, quello di creare e validare un nuovo sistema capace di investigare tramite FOT, i cambiamenti che avvengono a livello di meccanica polmonare, durante la rianimazione dei neonati. Il setup sviluppato sfrutta un ventilatore neonatale fornito da Acutronic’s: il Fabian’s HFO. Il modulo di alta frequenza del Fabian, HFOV è stato utilizzato come generatore di oscillazioni sinusoidali, come necessita la tecnica FOT. In accordo con la ditta Acutronic, alcune modifiche al firmware sono state fatte, incluso anche il trasferimento dei dati di flusso e pressione, tramite un preciso protocollo di comunicazione seriale. Altri cambiamenti software sono stati fatti. Per permettere ai care-giver di visualizzare on line la resistenza e la reattanza R ed X, senza cambiare eccessivamente il firmware del ventilatore, si è pensato ad un tablet android . Il tablet è un dispositivo estremamente utilizzato al giorno d'oggi, per l'estrema portabilità, il basso costo e per l'apprezzata filosofia del codice *open source*. Proprio grazie a queste caratteristiche, è stato possibile creare un'applicazione android ad hoc. L'applicazione registra i dati di pressione e flusso trasmessi dal ventilatore. Inoltre, calcola l'impedenza respiratoria Z_{RS}, scomposta nelle sue componenti reali ed immaginarie, R ed X. Le due misure registrate e queste ultime due derivate, sono plottate online, seppur sotto campionate, e salvate in un file per poter essere osservate successivamente. Il set-up creato ha il vantaggio di essere facile da usare e di essere compatto in quanto comporta solo un tablet ed un ventilatore. Pertanto può essere utilizzato in casi di emergenza, ad esempio nelle terapie intensive neonatali- NICU - o durante momenti critici come la nascita.

Test *in vitro* sono stati condotti per verificare la configurazione sviluppata. Dopo di che, il set-up è stato utilizzato *in vivo*. L'impedenza respiratoria è stata misurata in tre neonati in terapia intensiva neonatale della Mangiagalli e Regina Elena di Milano. I neonati sono stati studiati a livello basale con due setup: quello creato, ed uno già convalidato, tuttavia ingombrante e non utilizzabile nei momenti critici dei quali si è già parlato. Successivamente, la meccanica polmonare è stata studiata durante l'applicazione del SLI, solamente con il set-up sviluppato in questa tesi. Lo scopo di questo studio era di valutare la fattibilità del monitoraggio dei parametri di meccanica respiratoria durante la manovra SLI, in bambini nati pre termine.

Ciò di cui sopra, è stato mappato in questo mio elaborato di tesi, in 5 principali capitoli.

Il primo capitolo comprende la teoria inerente il sistema respiratorio completo e sviluppato (quello adulto), e le principali differenze tra questo e quello dei neonati. FOT è presentata come una tecnica non invasiva per monitorare le meccanica respiratoria. Dopo questi cenni teorici, sono spiegate la manovra di reclutamento SLI ed i principi di funzionamento dei ventilatori meccanici.

Il secondo capitolo invece tratta i principali strumenti utilizzati. Sono quindi descritti i componenti hardware ed il software del ventilatore Fabian. Il ventilatore è stato osservato ed è stata vagliata ogni possibilità di modifica, per creare il sistema di monitoraggio desiderato. Sono state trovate due possibilità per implementare le oscillazioni forzate FOT, che permetterebbero lo studio della meccanica polmonare dei soggetti: implementazione esterna al ventilatore, o interna ad esso.

Per la necessità di salvare spazio, specialmente in situazioni di emergenza, e per il desiderio di creare un singolo e compatto sistema, è stata scelta la strada dell'implementazione interna al ventilatore. Per cambiare il meno possibile il firmware del ventilatore, si è scelto di utilizzarlo nella modalità HFOV. Tale modalità di ventilazione con oscillazioni ad alta frequenza permette di fornire al paziente, un valore medio di pressione, al quale sono sovraimposte oscillazioni sinusoidali, con frequenza impostabile. Secondariamente, si è scelto un tablet per gestire i dati. Il tablet android sembra essere un'ottima scelta per il relativamente basso costo d'acquisto, l'alta portabilità e la possibilità di creare un'app apposita abbastanza facilmente, grazie ai codici open source reperibili in rete. Dunque, sono anche fatti accenni al linguaggio di programmazione ad oggetti, alla base della programmazione android.

Dopo aver preso le suddette decisioni, si è quindi creato il set-up. Nel terzo capitolo sono quindi descritti tutti i passaggi necessari per l'ottenimento del set-up clinico sviluppato. Il firmware del ventilatore di proprietà Acutronic è stato modificato in modo da inviare i dati di flusso e pressione percepiti prossimalmente al paziente, secondo un preciso pacchetto di dati e tramite protocollo seriale. Grazie all'applicazione android appositamente creata, il tablet controlla la ricezione dei dati e, in tempo reale, calcola le componenti reali ed immaginarie dell'impedenza respiratoria Z_{RS}. Il calcolo è possibile grazie all'abbassamento della soglia reimpostata sulla ricezione dei dati da parte del ventilatore. Tale soglia serve per eliminare eventuale rumore nei dati, tuttavia inficia la ricezione delle oscillazioni forzate necessarie per la FOT, la quale necessita di un'alta sensitività. È stata fondamentale anche la comprensione dello slittamento nel tempo dei dati di pressione e flusso, i quali quindi non risultavano perfettamente sincronizzati. Dopo aver compreso l'entità dell'errore, questo è stato compensato. In ultimo, il capitolo parla dell'applicazione android. Il tablet registra i dati inviati dal ventilatore, calcola le grandezze derivate di interesse, le plotta e le salva. Inoltre, viene applicata la compensazione che corregge lo slittamento nel tempo.

Il capitolo numero 4 tratta i test eseguiti in vitro. Alcune bottiglie e capillari in vetro, di grandezze note, costituiscono dei modelli polmonari (test lung). Ne sono stati creati 3, con diversi valori di resistenza e reattanza. Per trovare l'impedenza respiratoria Z_{RS}, è necessario che ci sia perfetta sincronia tra i dati di flusso, e quelli di pressione relativi allo stesso istante di campionamento. La compensazione di questo sfasamento nel tempo è stata trovata nel capitolo precedente. In questo quarto capitolo invece, la si è testata, grazie ai due rimanenti test polmonari. La validazione consiste nel comparare i valori ottenuti con il set-up clinico, creato per questa tesi, con quelli ottenuti dal set-up di riferimento, considerate come gold standard. La media degli errori in percentuale, tra le due misure è sotto il 10%, pertanto accettabile secondo le linee guida europee per la FOT:
Errore % Resistenza 4.37 ± 3.99 cmH_2O*s/l, Error % Reattanze 5.42 ± 4.21 cmH_2O*s/l.
Sono stati eseguiti anche altri test per verificare l'applicabilità della FOT durante la manovra SLI, e per verificare le performance del tablet. Più precisamente, per vedere se il tablet perde qualche dato, rispetto al gold standard, e se i dati sono processati nella maniera corretta, anche se i calcoli vengono eseguiti run time e non successivamente.

Il quinto capitolo riguarda la validazione clinica ed i risultati. Per verificare che il set-up sviluppato sia del tutto affidabile, sarebbe infatti necessario testarlo in condizioni critiche e, certamente, non ideali. Quindi dei neonati sono stati osservati alla clinica Mangiagalli di Milano. I pazienti sono stati scelti dal personale medico: età gestazionale media 31.33 ± 2.3 settimane, età dopo la nascità: 18 ± 10.3 giorni e peso medio alla nascita: 1195 ± 169 g. Il confronto tra i valori ottenuti nelle due modalità di acquisizione ormai note, è stato fatto sui tratti basali, ovvero senza insufflazione di alte pressioni, né l'applicazione di particolare manovre. Si è quindi potuti procedere con lo studio durante l'applicazione della manovra SLI.
In questo caso, la manovra SLI è stata applicata unicamente con il set-up clinico. Si son considerati diversi momenti temporali ed i risultati sono elencati e commentati direttamente nello stesso quinto capitolo. Su 3 neonati studiati, la SLI si è dimostrata non

essere sempre soddisfacente. Ciò dipende, probabilmente, dalle caratteristiche polmonari del singolo individuo. È quindi possibile verificare, tramite il set-up creato, quali pazienti necessitino o meno di reclutamento, ed in quale entità.

In conclusione, il setup sviluppato è:

- permette misure accurate di resistenza e reattanza durante la manovre SLI, applicata su neonati pretermine
- il set-up non è ingombrante, ma compatto
- il set-up è user friendly
- il set-up e la procedura sono ben tollerati dai pazienti

Il sistema creato può essere utilizzato per eseguire successivi studi, i quali scopi saranno la valutazione dei benefici clinici portati dalla scelta dei parametri di reclutamento polmonare.
Nel futuro si auspica una modifica del set-up attuale, se non altro per vedere quale configurazione si rivela essere la migliore; la generazione delle oscillazioni forzate, necessarie per la FOT, potrebbe essere gestita quindi da un controllo sulla valvola PEEP e non più tramite il modulo HFO.
Un ulteriore sviluppo, potrebbe riguardare il software android. Questo, automaticamente potrebbe riconoscere i tratti più stabili di resistenza e pressione, evitando gli spike dovuti ad eccessive perdite o ad eccessiva opposizione respiratoria da parte del bambino. Il riconoscimento del tratto più stabile di fine espirazione dovrebbe avvenire non solo tramite l'osservazione del flusso respiratorio, ma anche del volume tidalico espiratorio.
Prossimi studi, in ultimo, potrebbero essere indirizzati sulla scelta della frequenza di stimolazione ottimale. Basse frequenze sono ottimali per l'analisi delle zone periferiche polmonari, tuttavia il SNR – rapporto segnale rumore è basso a causa delle interferenze con le componenti del segnale respiratorio. Negli adulti 5 Hz è il compromesso ideale, mentre nei neonati che hanno frequenze respiratorie nettamente maggiori ed una più alta frequenza di risonanza, la scelta di una frequenza di stimolo maggiore potrebbe essere l'ideale.

INTRODUCTION

The respiratory system conducts and exchanges gas with the pulmonary circulation. Adult respiration is very different from newborns' one, because when infants are born, the respiratory system is not completely developed yet – particularly when birth occurs preterm [1, 2]. These differences result in more a higher susceptibility to respiratory stress. Birth represents a critical event associated with dramatic changes in lung function: the fetal lung fluid which maintains the lungs in a distended state during fetal life needs to be absorbed, the lungs have to be filled with air, adequate gas-exchanging surface area has to be established, the pulmonary blood flow has to greatly increase and the surfactant system has to ensure the lungs remain expanded by decreasing the alveolar surface tension [3].

All these features make infants and especially the preterm ones very exposed to respiratory system pathologies [4]. Ten to fifteen percent of all newborn babies require care in NICUs, specialized Intensive Care Units for Neonates. For the onset of ventilation the respiratory muscles have to provide an adequate force to overcome the elastic recoil of lungs and chest wall, the viscous friction of air flowing and the forces caused by the viscoelastic behaviour of the lung tissues and the inertial forces.

When, for any reason, the respiratory muscles cannot provide the necessary pressure for respiratory flow, ventilation must be guaranteed externally, by artificial ventilators. Assisted ventilation aims at re-expanding atelectatic lung units and at preventing the alveoli from collapsing, resulting in increased gas-exchange surface and in improved ventilation-perfusion matching. Although mechanical ventilation is indispensable for the survival of little patients, excessive tidal volumes and inadequate lung recruitment may contribute to mortality by causing ventilator-induced lung injury (VILI).

Nowadays, recruiting strategies are investigated in order to assist newborns right at birth helping them to remove the fetal fluid from the lung and establish an adequate volume recruitment, as alveolar recruitment could prevent mechanical ventilation and thus the secondary derived injury, or at least to reduce the ventilator induced lung injury. A lot of techniques have been proposed to recruit the lung. One of them, the Sustained Lung Inflation (SLI), a technique which insufflates air at high pressure peaks for few seconds, is gaining popularity [5, 6]. SLI has been proved to be an effective way of recruiting, anyway the knowledge of respiratory mechanics during this manoeuvre is fundamental in order to define an optimal approach to maximize the final functional residual capacity (FRC) and to minimize the stress applied to pulmonary tissue [6, 7].

SLI has been proved to be more effective than Intermittent Mandatory Ventilation (IMV) for term newborns, according to the FRC with asphyxia. Moreover, recent studies tested the good scores achieved even in pre-term newborns. This recruitment aims at achieving, after the inflation, to a natural respiration [6].

The need of a monitoring system, able to observe these changes in lung mechanics, is strong. In the '50s DuBois et al. proposed a method to calculate the respiratory impedance, due to the elastic recoil of lung and chest wall, the viscous friction of air flowing and the forces caused by the viscoelastic behavior of the lung tissues and the inertial forces. This methods, called Forced Oscillation Technique (FOT), permits to obtain the input impedance of the respiratory system, a complex number, which accounts for gas compression, airways and tissue impedance. In particular, the imaginary part of the impedance represents the compliance and the inertance of the overall respiratory system and is called reactance, X_{RS}, while the real part represents the resistance of the overall respiratory system, R_{RS}. FOT is performed by measuring respiratory system's response to a forcing signal applied externally [8, 9, 10]. It is a simple and non-invasive technique and it does not require patient's collaboration and therefore it is very useful to study infants who are not capable to direct collaboration. Particularly, it has been recently shown that X_{RS} measured at 5Hz – at least in adults - is very sensible to changes in the mechanics of lung periphery and provides accurate information about lung volume recruitment and derecruitment. The above mentioned Forced Oscillation Technique could be useful to measure newborns' mechanical properties without imposing a direct collaboration [11].

It should be very useful to have a unique device to measure and stimulate the subjects, in order to reduce set-up in NICUs and delivery rooms, where there are already a lot of instruments and space needs to be saved. In this way it should be possible to ventilate subjects, observing even mechanical parameters such as the respiratory impedance.

The main purpose of this work is thus to develop a setup which is able to monitor changes in lung mechanics, during the application of SLI resuscitating manoeuvre, by FOT. The

setup has been developed with forced oscillations implemented by the ventilator. An android tablet works with the ventilator, and processes data delivered by it in order to calculate resistance and reactance values, which are saved and on line displayed too.
The developed system has been tested both in vitro e in vivo.

Introduction References

1. Horsfild, K., G. Dart, D. E. Olson, and G. Cumming. Models of the human bronchial tree. J Appl Physiol. 1971;31:207-17.

2. Merkus PJ, ten Have-Opbroek AA, Quanjer PH. Human lung growth: A review. Pediatr Pulmonol. 1996;21(6):26-36.

3. Guyton AC HJ. Il sistema respiratorio. In: Fisiologia Medica. ; 2006.

4. Garvey J. Infant respiratory distress syndrome. Am J of Nursing. 1975;75(4):614.

5. Dellaca RL, Andersson OM, Zannin E, Kostic P, Pompilio PP, Hedenstierna G et al.: Lung recruitment assessed by total respiratory system input reactance. Intensive Care Med 2009.

6. Harling E, Beresford MW, Yoxall CW. Does sustained lung inflation at resuscitation reduce lung injury in the preterm infant?. Arch Dis Child. 2005;90(5):406-10.

7. te Pas AB, siew M, Yagi N et al. Establishing functional residual capacity at birth: The effect of sustained inflation and positive end-expiratory pressur in rabbit lung model.. Pediatric Research. 2009;65(5):597-41.

8. Oostveen E, MacLeod D, Lorino H, Farre R, Hantos Z, Desager K, Marchal F, ERS. The forced oscillation technique in clinical practice: methodology, recommendations and future developments. Eur Respir J. 2003;22(6):1026-41.

9. Dellaca RL, Veneroni C, Vendettuoli V, Zannin E, Matassa P, Pedotti A, Colnaghi M, Mosca F In press, Relationship between respiratory impedance and positive end-expiratory pressure in mechanically ventilated neonates Intensive Care Med

10. Gauthier R, Beyaert C, Feillet F, Peslin R, Monin P, Marchal F. Respiratory oscillation mechanics in infants with bronchiolitis during mechanical ventilation. Pediatr Pulmonol. 1998;25(1):18-31.

11. Weber K, Courtney SE, Pyon KH, Chang GY, Pandit PB, Habib RH. Detecting lung overdistention in newborns treated with high-frequency oscillatory ventilation. J Appl Physiol. 2000;89(1):364-72.

1 - THEORETICAL BASES

This chapter deals with the fundamental of respiratory system, the characteristics of the preterm newborns’ respiratory system, the recruitment maneuvers and the mechanical ventilator functioning principles.

Lung ventilation is the first step for respiration and it is fundamental to provide the correct amount of oxygen to the tissues. Infant respiratory system has several physiological and anatomical differences with respect to the adults' one. Moreover, preterm infants are compounded with an incomplete lung structure. When, for any reason, the respiratory muscles cannot provide the necessary pressure for respiratory flow, the lung ventilation must be guaranteed externally, by artificial ventilators. Anyway excessive tidal volumes and inadequate lung recruitment may contribute to mortality by causing Ventilator Induced Lung Injury – *VILI*. The recruitment of alveoli is necessary to reduce lung injury. SLI – Sustained Lung Inflation - is a recruiting technique which insufflates air at high pressure peaks for few seconds that is entering the clinical practice especially for recruiting lung immediately after birth. The knowledge of what happens to lung tissues during these moments, will help to define an optimal approach to maximize the final FRC and to minimize the stress applied to pulmonary tissue. Up to now there are no methods to monitor this maneuvers, however the forced oscillation technique (FOT) is a non-invasive and versatile technique to assess respiratory mechanic, without patient’s cooperation, which can be implemented in ventilator and can be used for this purpose.

1.1 Physiology and Mechanics of Breathing

1.1.1 The Respiratory System

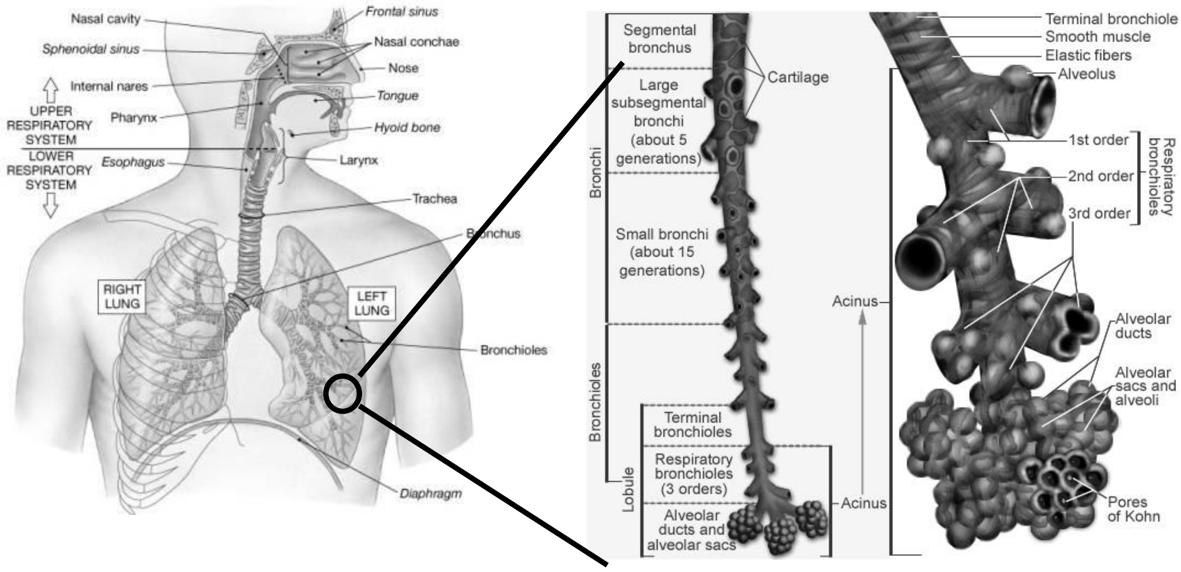


Figure 1. 1 Global view of respiratory system (left) and zoomed view of bronchial tract (right)

The developed respiratory system is the human system which conducts and exchanges gas with the pulmonary circulation. It is composed by different anatomical parts. From the external body, moving towards the internal one, there are mouth and nose, upper and lower airways.[1] The former airways consist of sinuses, vocal cords and larynx; the latter instead of trachea, bronchi, bronchioles. [2, 3]

The trachea divides into two bronchi, nominated “right“ and “left”. They can be furthermore divided in bronchioles. Although gas exchanges starts in the terminal zone of bronchioles, alveoli are the active part of exchange of gas. Since there are no alveoli in this initial part of airways (conducting airways), it is defined anatomic dead space (about 0.15 l). The “respiratory zone” is so composed of bronchioles, which have occasional alveoli on their walls, and of alveolar ducts which are completely filled with alveoli. The alveolus is the air-blood interface and allows gas exchange by diffusion. [4] There are about 300 million alveoli, about 0,3 mm in diameter each. Anyway, the respiratory system is not only composed of the lung but also by the chest wall. The pleural liquid, a thin liquid film at sub-atmospheric pressure, keeps lungs and ribcage close to each other. The interaction between these two structures determines the lung volume which plays a fundamental role in gas exchange and in work of breathing.

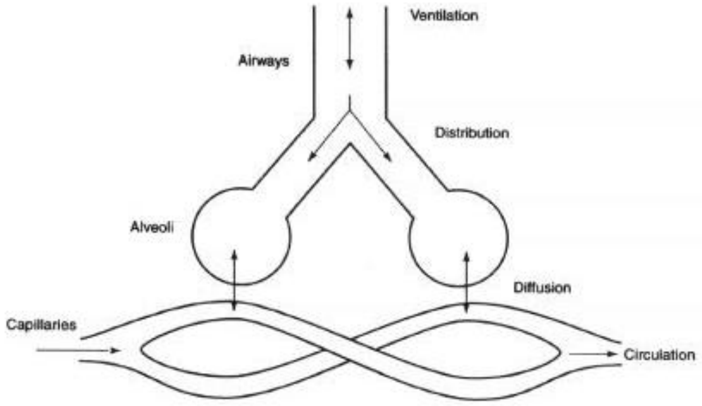


Figure 1. 2 Alveolar gas exchange

1.1.2 Definition of Lung Volumes and Pressures

A simple method for studying pulmonary ventilation is to record the volume movement of air into and out of the lungs: the spirometry. It is possible to define lung volumes under different conditions – as showed in Figure 1. 3: [4,5]

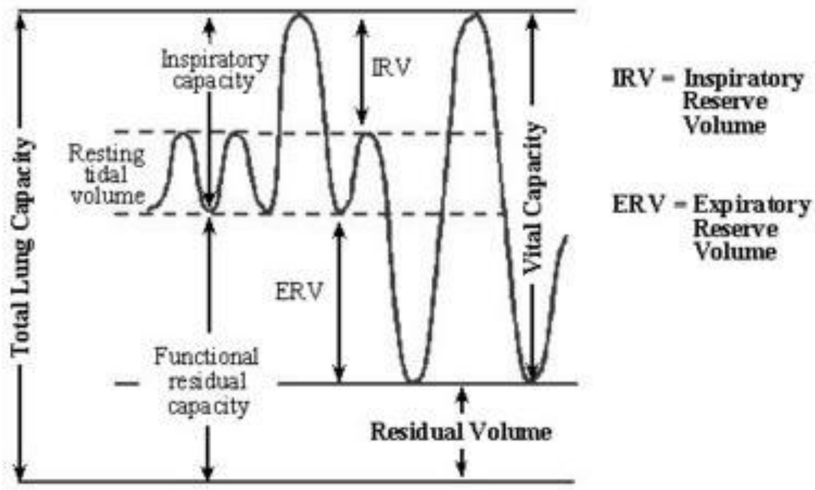


Figure 1. 3 Definition of lung volumes

- Tidal Volume (VT) is the volume of a spontaneous breath (about 500 ml).
- Inspiratory Reserve Volume (IRV) is the maximum volume inspirable from an end-tidal inspiratory level.
- Expiratory Reserve Volume (ERV) is the maximum volume that can be expired from the end-expiratory level.
- Residual Volume (RV) is the volume of gas in the lungs and airways after as much gas as possible has been exhaled (about 1.2 l).

- Total Lung Capacity (TLC) is the volume of gas in the lungs and airways after as much gas as possible has been inhaled (about 6 l).
- Vital Capacity (VC) is the volume of the deepest breath.
- Functional Residual Capacity (FRC) is the volume of gas in the lungs and airways either at the end of spontaneous expiration or at the resting volume of the respiratory system.
- Inspiratory Capacity (IC) is the maximum volume that can be inspired from the end-expiratory level.

To best compare lung volumes among subjects, generally these variables are corrected expressing them in percent of VC. This method is very efficient also because in normal subjects of equal sex, habits and age these values may reflect noticeable differences. Changes in lung volume are the result of the air flow in and out the respiratory system. Convective flow occurs as a result of a difference in pressure between two points. The respiratory system is able to establish a difference in pressure between open air and alveolar space by dimensional variations of thoracic cavity, generated thanks to the contraction and the relaxation of the chest muscles. Thus different following pressure can be defined:

- External Pressure or Pressure at the Body Surface (P_{BS})
- Pressure at the Opening Airways (P_{AO}) is the pressure measured at the mouth.
- Alveolar Pressure (P_{ALV}): the one registered within the alveoli.
- Pleural Pressure (P_{PL}) refers to the pressure within the pleural liquid that in normal condition is sub-atmospheric.
- Transpulmonary Pressure (P_{TR}): it is defined as the difference between the alveolar pressure and the pleural pressure. It is a measure of elastic forces that tend to collapse the lung at any level of lung expansion.
- Wall Pressure (P_W) results from the difference between the pleural pressure and the external pressure.
- Respiratory System Pressure (P_{RS}) is the pressure to which the respiratory system is exposed. It is calculated as the sum of the pressure drop due to airways (P_{AW}, the difference between P_{AO} and P_{ALV}), the transpulmonary pressure and the wall pressure. P_{RS} can be divided in a resistive component (P_{RES}), due to flow and to

dissipation properties of the system, and in two elastic components, referred to lung ($_L$-P_{EL}) and to chest wall ($_{CW}$-P_{EL}), due to volume and conservative properties of the system. [6]

1.1.3 Lung Ventilation

The lung ventilation is performed let flowing air into and out of the lungs, in alternation.

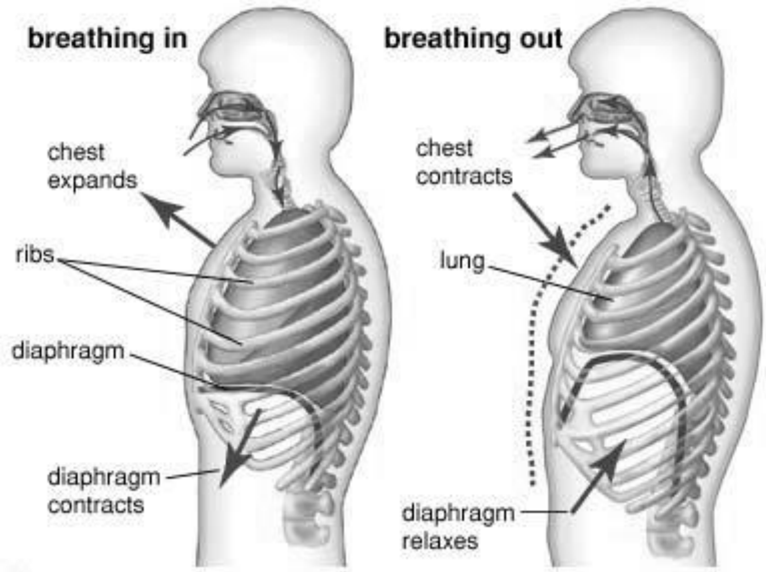


Figure 1. 4 Mechanic of inspiratory and expiratory phases

Inspiration

One of the two above mentioned phases, is the inspiration. Air flows in the lung when atmospheric pressure overcomes the pressure in the alveoli. Pressure can be generated both by the diaphragm and the intercostal muscles. When diaphragm contracts, abdominal pressure is increased and this generates forces able to elevate and enlarge the power part of the ribcage. If the action effectively transmits to the upper part, lung volume changes and inspiration can continues.

This transmission of actions depends on the compliance of the rib cage: the less it is compliant, the more the risk to have paradoxical breathing. External intercostal muscles helps the rotation of the ribcage forward and upward. Moreover, it pushes the sternum increasing the antero-posterior and cross-section diameter of the thoracic cage and producing negative pressure in the intrapleural space. As before said, the increasing of the chest volume makes the lungs expanded because of the cohesive forces between the visceral pleura and the parietal one; the lungs and their inner pressure decrease letting the atmospheric air enter alveoli. [4, 6]

Expiration

The second phase is usually passive, at least in quiet breathing. It deals as the inverse of the physical phenomena which provoke the inspiration. It is driven by the elastic recoil of the chest and of the lungs which were stretched during inspiration. It depends on surface tension, elastic properties of the tissues and bony elements of the rib-cage too. Active expiration instead is determined by the muscles of the abdominal wall and by the internal intercostals. When muscles of the abdominal wall contract, the abdominal pressure raises pushing the diaphragm upward. The thoracic volume decreases. [4, 6]

1.1.4 Mechanical Behavior of the Respiratory System

The mechanical behavior of the respiratory system can be explained using an electrical equivalent model - Figure 1. 5. It concerns four elements: airways, lung, chest wall and respiratory muscles. As usual, electric charge is the volume and thus the derivate, the electric current represents the flow of air; electric potential stands on the other hand for the pressure. The respiratory muscles are represented as voltage generator because they are able to change pleural pressure (P_{PL}) through the chest wall (CW). Because of compression and expansion of gas, a capacitor is inserted between the airways (AW) and the lung (L) – the pressure between airway and lung is the alveolar pressure, P_{ALV}. The equivalent describes the behavior of lungs which neither suddenly expand during expiration nor decrease in volume when inspiration starts. Capacitor manages the usual exponential increment and decrement.[5] P_{BS} is the pressure at body surface, while $_{AO}$ is the acronym for airway open. The gradient is P_{RS}, respiratory pressure.

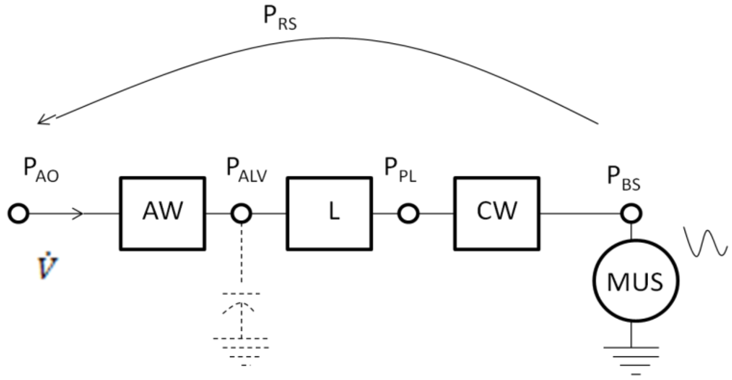


Figure 1. 5 Mechanical equivalent

Static Behavior

Volume-Pressure Curve of the Lung

The lung is an elastic structure which would collapse expelling all its air through trachea if there were no force that keeps the air inflated. Lung tissue has fibers of elastin and collagen in it.

Pressure-Volume curves represent the behavior of lung or of the total respiratory system in static or dynamic condition. The static P-V curve is assessed by plotting the volume of the lung versus the driving pressure: the slope of this curve is known as compliance and describes the distensibility of the lung. Static P V curve can be observed in Figure 1. 6: two borders are defined as HIP and LIP as explained in the caption.

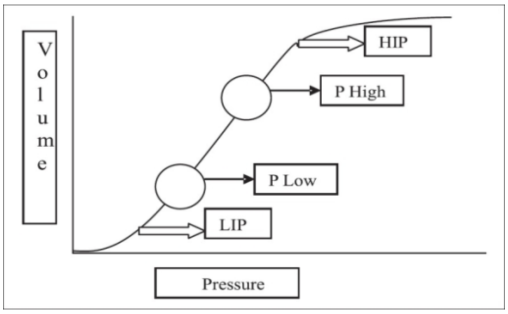


Figure 1. 6 Static P V curve. High and low pressure values are decided in the range of HIP –high inflection point- and LIP – low infection point

The steeper the slope, the grater the compliance and then the distensibility of the lung. Considering the whole breathing, the curve is hysteretic, namely it has two different P-V relations for expiration and inspiration [7], as depicted in Figure 1. 7:

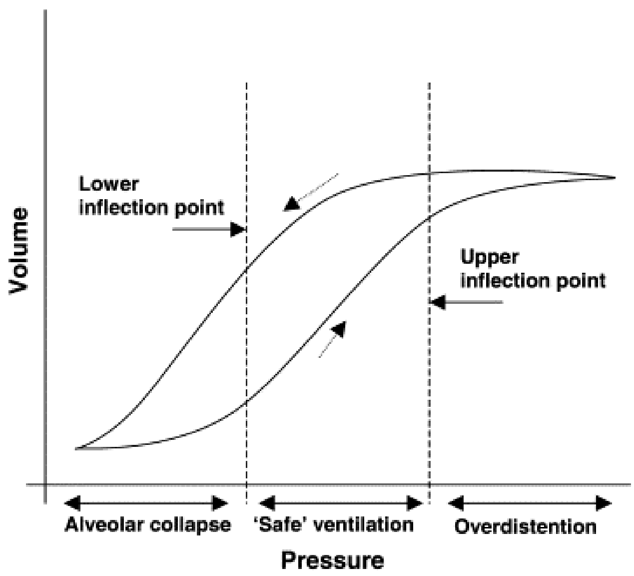


Figure 1. 7 Hysteric dynamic P-V curve. A tract corresponds to inspiration while B to expiration phase. There are three zones where alveoli behave in different ways.

The curve deserves to be analyzed at least in 3 zones: the central and the two external regions. The lower end of the curve has a low compliance. This region is below critical opening pressure so pressure is required to open terminal airways and alveoli. At the centre of the curve the compliance increases resulting in a large change in volume for a small change in pressure. Normal tidal breathing usually occurs here because efficiency is maximal. At the upper end of the curve the compliance decreases once more. At high expanding pressures, in fact, the lung is overinflated and opposes to further volume increases. [8]

Lung's major elastic component is probably due to the geometrical disposition of these fibers. [3, 6] Furthermore, the compressibility of the gas and the surface tension at the alveoli cause elastic behavior too. However the surface tension at the air-alveolus interface that allows mechanical equilibrium is really lower than in the presence of air-water interface. In fact it is highly dependent on the presence of the surfactant which reduces surface tension at the alveolar air-liquid interface. Surfactant is a complex lipoprotein formed by type II alveolar cells that lower the surface tension and consequently reduce the work of inspiratory muscles. This material is adsorbed at alveolar interface and its effect depends on alveolar size: the smaller the alveolus' radius the more concentrated is the surfactant and this results in a further reduction of the tension. [7, 9] Surfactant helps to keep the alveoli dry. In fact, the surface tension tends to collapse alveoli and to adsorb fluid into the alveolar space from the capillaries. The surface tension of alveoli reduces the hydrostatic pressure outside the capillaries by reducing surface forces; surfactant prevents the transudation of fluid into the interstitium. Anyway, all these components concur in the definition of elastic nature and a curve testifies a scientific and proportional correlation between the driving pressure and the change in volume.

An important factor in the pressure-volume behavior of the lung is, as said, the surface tension of the liquid film covering the alveoli. It is a contractive tendency of the surface of a liquid that allows it to resist an external force. To be more explicit, surface tension arises because the attractive forces between adjacent molecules of the liquid are stronger than those between the liquid and gas and thus, it keeps occupying the smallest possible area. It is revealed, for example, in the ability of some insects to run on the water surface. This property is caused by cohesion of similar molecules, and it is responsible of many of the behaviors of liquids. The tendency to occupy the smallest possible area, brings to make a sphere-like form. The pressure within a liquid sphere should be influenced both by the surface tension forces offered by the liquid and by the size of the sphere. Indeed, Laplace, found that pressure within a sphere is directly related to the surface tension of the liquid and inversely related to the radius of the sphere, or

$$P = 2\frac{\gamma}{R}$$

[Eq. 1. 1]

where P is the pressure within the sphere, γ is the surface tension of the liquid , and R is the radius of the sphere – make reference to Figure 1. 8. As equal, thinking the alveolus as a spherical cavity

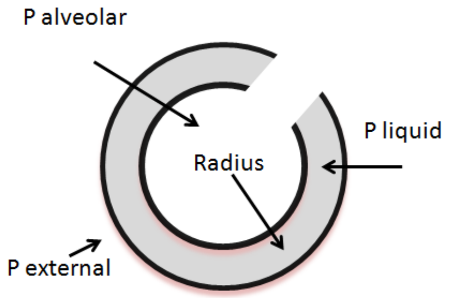


Figure 1. 8 Spherical alveolus

with a thin wall covered by a liquid film, the total pressure across the alveolar wall (P) is the sum of the pressure gradient at the air liquid-interface (P_L-P_{EXT}) and the pressure between the liquid film and the alveolar wall (P_{ALV}-P_L):

$$P = P_{ALV} - P_L + P_L - P_{EXT} = \frac{2\gamma}{R} + \frac{2\gamma}{R} = \frac{4\gamma}{R}$$

[Eq. 1. 2]

where γ is the surface tension and *R* the radius of the sphere ([Eq. 1. 1] previously written) This relationship shows that the pressure (*P*) needed to stabilize the alveolus is directly proportional to the surface tension (*γ*) at air-liquid interface and inversely proportional to the radius (*R*). So the larger the radius of the alveoli, the less pressure is needed to hold them open, the smaller the radius, the more pressure is required to hold the airways open. [6]

Volume-Pressure Curve of the Entire Respiratory System

It has been spoken about lungs and their tendency to collapse; the main force which opposes the collapse is the one provided by the rib cage. The curve of the global mechanical answer of the respiratory system corresponds so to the sum of the curve of lung and chest wall measured separately, using the principle of superposition of effects. The chest wall curve alone, has negative pressure at low volumes so as at FRC and becomes positive when the volume is increased to about 75% of the vital capacity (VC). In the lung curve alone, pressure is always positive and the lung tends to collapse because of its elastic properties. At volumes above FRC, the total pressure of lung and rib cage together is positive and at smaller volumes the pressure becomes sub-atmospheric. At zero pressure, the lung is at its minimal volume, which is below residual volume (RV), the minimal volume reached by the chest wall only. [7, 10]

P-V curves have great importance in deciding the optimal parameters to recruit lungs, as it will be discussed in the next paragraph.

Dynamic Behavior

During respiration the respiratory muscles have to provide an adequate force to overcome the elastic recoil of lung and chest wall, the viscous friction of air flowing through the airways, the forces caused by the viscoelastic behavior of the lung tissues and the inertial forces. In expressing this, it is not used an electrical model, but a mechanical one is preferred. The balance of forces is described by equation [Eq. 1. 3,which is known as the equation of motion of the respiratory system.

$$P_{AO} = \frac{V}{C} + \dot{V}R + \ddot{V}I - P_{MUS}$$

[Eq. 1. 3]

In this equation P_{AO} is the pressure at the airway opening, while V, $V^{\cdot}$ and $V^{\cdot\cdot}$ are the volume of the lung above end-expiratory volume and its time derivatives, flow and acceleration. P_{MUS} is the pressure generated by the respiratory muscles. Other fundamental parameters are compliance *C*, resistance *R* and inertance *I*. The parameters relates pressure to Volume, flow and acceleration respectively.

Compliance

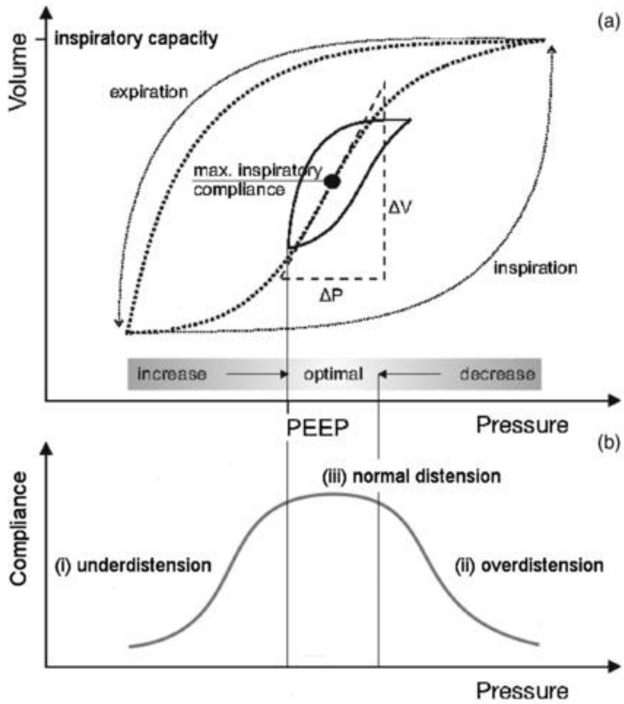


Figure 1. 9 Dynamic P V curve (upper) and related compliance (lower).

In the optimal zone, the compliance assumes the highest and smoothest value.

Compliance measures the elastic recoil of the lung and chest wall. It depends primarily on lung's size in a direct proportional way. [4, 6]

$$C = \frac{\Delta V}{\Delta P}$$

[Eq. 1. 4]

Elastance is a very similar way to refer compliance; it is exactly the inverse of C. Static and dynamic compliance can be distinguished. Static compliance is obtained considering the slope of the static P-V curve while dynamic compliance is related to the dynamic P-V curve. Dynamic pressure-volume loops show the pressure-volume relationship during inspiration and expiration (make reference to Figure 1. 9).

At end inspiration flow is zero, but pressure is lower than that expected from the static curve. The difference between static and dynamic compliance is due to resistance heterogeneities in the alveolar units of the same generation. If the airway diameter is smaller, flow is reduced and volume changes are slower. As frequency increases, if alveoli have not enough time to empty, the total change in lung volume decreases and therefore also compliance decreases. [6]

Resistance

Resistance represents the mechanical opposition to flow because of frictional forces. The resistance of the respiratory system is the sum of viscous resistance and airways resistance. Viscous resistance is the resistance generated within the lung tissue and chest wall during inflation and deflation. [6] Airway resistance is defined as the pressure gradient needed to move gas though the airways at a constant flow rate:

$$R = \frac{\Delta P}{\dot{V}}$$

[Eq. 1. 5]

R not only depends on length and inside diameter of the conducting airways, but also on viscosity and density of the gas, on flow rate and on the kind of flow, which can be laminar or turbulent. Reynolds' number is used to characterize these regimes of flow. Flow through straight tubes starts to become turbulent when Reynolds' number exceeds approximately 1500 and is generally considered to be fully turbulent when Reynolds' number is above 4500. [11] It has the following expression:

$$Re = \frac{\nu \rho ID}{\mu}$$

[Eq. 1. 6]

where v is fluid velocity, ρ gas density, ID is the internal diameter of the tube and μ is the viscosity of the gas. When both the flow regimes are presents the relationship between pressure and flow is expressed by Rohrer's equation:

$$\Delta P = K_1 \dot{V} + K_2 \ddot{V}^2$$

[Eq. 1. 7]

where K_1 and K_2 are called the Rohrer constants. When flow is fully turbulent, the pressure difference is proportional to the square of the gas, only the second part of equation has to be considered and the resistance coincides whit K_2; while when flow is fully laminar only the linear part of equation has to be considered and the resistance coincides whit K_1.

The resistance of the respiratory system (R_{RS}) represents the sum of airways, lung tissue and chest wall resistance and it is rarely linear so the Rohrer's equation is to be considered. Thus the human lung can be modeled as a multiple tube system whose resistance depends on the total cross-sectional area of all the tubes. (Figure 1. 10) Because resistance increases with the fourth power as the airway is narrowed, even small airway constrictions can cause significant increases in resistance to flow.

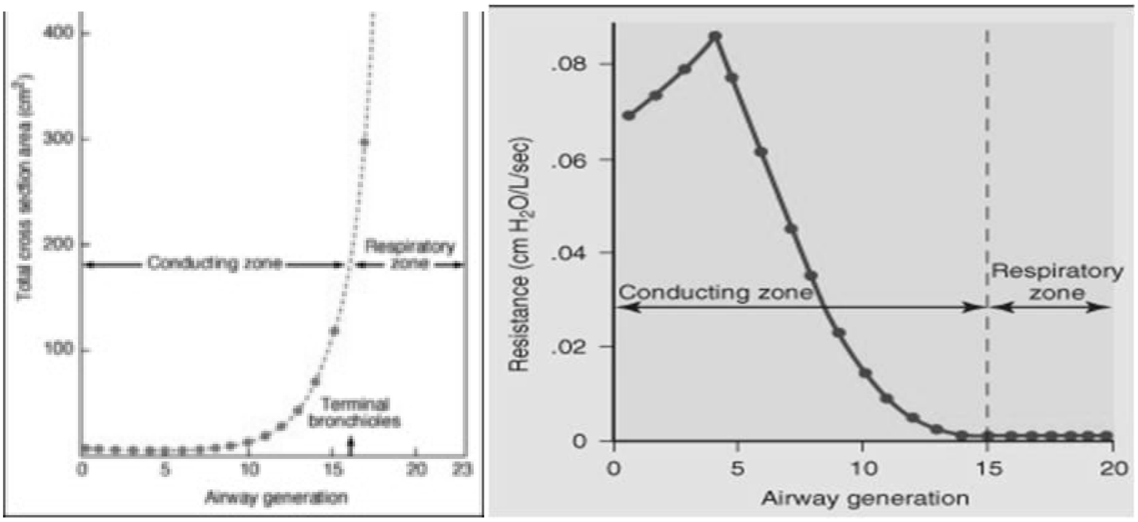


Figure 1. 10 Total cross sectional area (left) and resistance (right) values, as function of air way generation

Inertance

Inertance of the respiratory system is the analogue of the electric inductance and it is a measurement of the tendency of the respiratory system to resist changes in flow. At frequencies normally reached, during spontaneous and mechanical ventilation, inertance is usually negligible and in general ignored. The pressure caused by the inertance is in the opposite direction to that generated by the elastance. Therefore, inertial forces tend to balance the impedance to the flow provided by the stiffness of the respiratory system (make reference to [Eq. 1. 13] in the next paragraph). [6]

1.1.5 Oscillatory Mechanics and FOT

Oscillatory mechanic of the respiratory system is the study of the mechanical properties of the lung and airways as derived by the response of the system to small pressure stimuli generated externally and applied during normal breathing. The relationship between pressure and flow is non linear in the respiratory system but we can consider its linearization around a work point in order to analyze the behavior of the system using tools for studying linear systems. In fact in this way it is possible to define the transfer function between two variables, one chosen as input and the other one as output and to use it to evaluate the mechanical properties of the respiratory system. When considering pressure (P) as output and flow ($\dot{V}$) as input we obtain the respiratory impedance Z_{RS}, or input impedance Z_{IN}:

$$Z_{RS}(w) = \frac{P(t)}{\dot{V}(t)} = \frac{Pe^{(jwt+\varphi)}}{\dot{V}e^{(jwt)}} = |Z_{RS}|e^{j\varphi}$$

[Eq. 1. 8]

The respiratory system can be modeled as a generalized two-port system in which one port is the airways opening and the other one the body surface. For a two port system like that two different mechanical impedances can be defined.

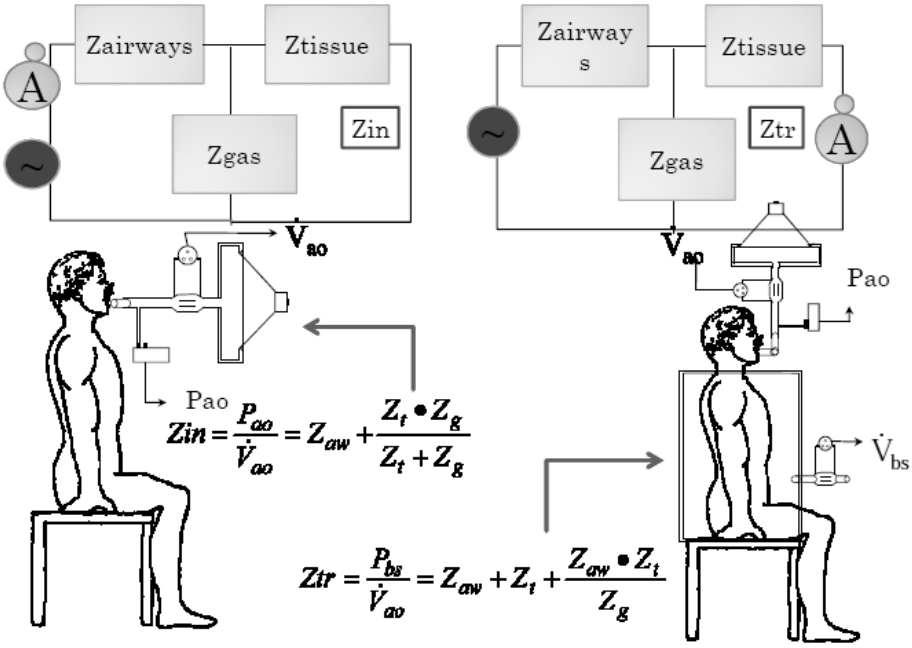


Figure 1. 11 Schematics of Ztr at chest or at mouth forced.

The input impedance of the respiratory system (Z_{IN}) is then obtained as the spectral relationship between pressure and flow (defined positive if entering the port) both measured at the airways opening when the stimulus is imposed to the same side, while the transfer impedance (Z_{TR}) is obtained when the oscillations are imposed and pressure and flow are measured at different sites of the respiratory system. The expression for Z_{IN} and Z_{TR} are the following:

$$Z_{IN}(f) = \frac{P_{RS}(f)}{\dot{V}_{AO}(f)}$$

[Eq. 1. 9]

$$Z_{TR} = \frac{P_{AO}}{\dot{V}_{BS}} = \frac{P_{BS}}{\dot{V}_{AO}}$$

[Eq. 1. 10]

where P_{AO} and $\dot{V}_{AO}$ are the pressure and flow measured at the airways opening respectively, while P_{BS} and V_{BS} are the pressure and flow measured the body surface, respectively. Since Z_{IN} and these two variants of Z_{TR} are affected differently by the parallel elements of the respiratory system, such as alveolar gas compressibility and upper air way wall movements, they can be selected or combined to obtain more reliable estimates of the airway and tissue impedance. If the measured system is linear, the two expressions above for Z_{TR} must be equivalent [56]. The internal connection between the two ports can be modeled using the T-network, proposed by DuBois et al. 41] (Figure 1. 12)

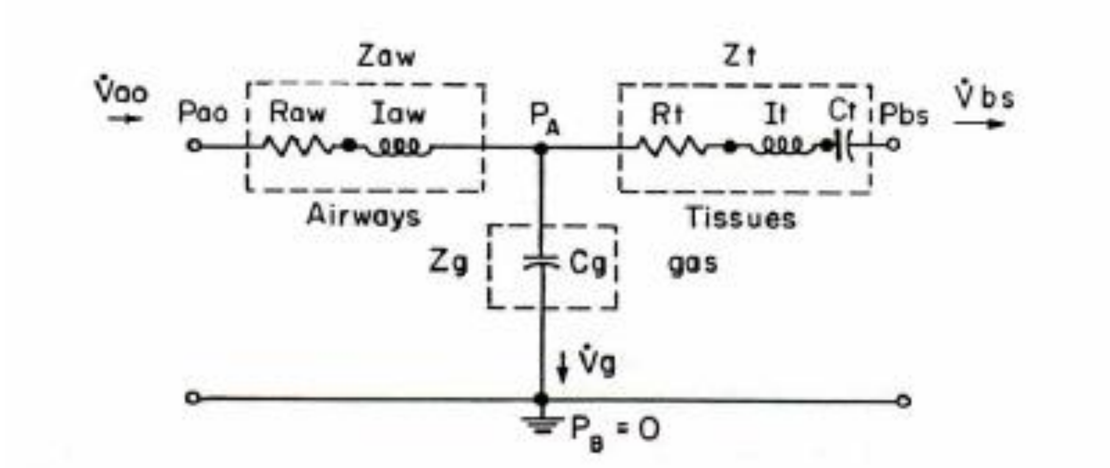


Figure 1. 12 Two-port equivalent system

It includes airways impedance (Z_{AW}), tissue impedance (Z_T) and gas compression (Z_G). Z_{AW} is composed of the correspondent resistance and reactance, capacitor element (C_G) accounts for gas compressibility and the tissue properties are described by the relative resistance, compliance and reactance. If alveolar gas compressibility is small ($C_G \approx 0$ and $\cdot V_G \approx 0$), then this model reduces to the series R-I-C model, where $R_{RS}=R_{AW}+R_T$, $I_{RS}=I_{AW}+I_T$ and $C_{RS}=C_T$. The electrical analogue can be simplified as it is possible to express the system as a one-port system:

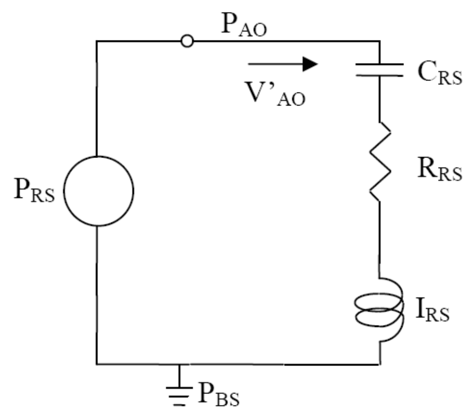


Figure 1. 13 One-port equivalent system

The total impedance of this simplified model of the respiratory system can be calculated as following, summing partial pressure drops:

$$P_{RS} = P_{AO} - P_{BS}$$

[Eq. 1. 11]

$$P_{RS} = \dot{V}_{AO} * R_{RS} + \dot{V}_{AO} * \frac{1}{j\omega C_{RS}} + \ddot{V}_{AO} * j\omega I_{RS}$$

Eq. 1. 12]

considering that PBS is the atmospheric pressure results:

$$Z_{RS} = \frac{P_{AO}}{\dot{V}_{AO}} = R_{RS} + j(\omega I_{RS} - \frac{1}{\omega C_{RS}})$$

[Eq. 1. 13]

The imaginary part of the impedance represents the compliance and the inertance of the overall respiratory system and is called reactance, $X_{RS}(\omega)$, while the real part represents the resistance of the overall respiratory system, $R_{RS}(\omega)$. Another way to consider respiratory impedance, is to divide its magnitude and its phase, as showed in [Eq. 1. 8]:

$$Z_{RS} = |Z_{RS}|e^{J\varphi_{RS}}$$

Where $|Z_{RS}|$ is the magnitude of respiratory impedance and Φ_{RS} represents its phase:

$$|Z_{RS}| = \sqrt{{R_{RS}}^2 + {X_{RS}}^2} \qquad \varphi_{RS} = \tan^{-1}\frac{X_{RS}}{R_{RS}} \qquad \text{[Eq. 1. 14]}$$

At low frequencies, because $1/\omega C_{RS}$ is greater than ωI_{RS}, the system behaves like a compliant one, the impedance magnitude decrease with frequency, Φ_{RS} is negative (pressure lags flow) and the imaginary part is negative. At higher frequencies, because ωI_{RS} is larger than $1/\omega C_{RS}$, the effect of inertance is predominant, the impedance magnitude increases with frequencies, Φ_{RS} is positive and the imaginary part is positive. At an intermediate frequency called resonant frequency (f_0), the effects of compliance and inertance cancel each other so that $|Z_{RS}|=R_{RS}$ and $\Phi_{RS} = 0$ (pressure in phase with flow).

FOT or Forced Oscillations Technique makes reference to this theory to gain respiratory impedance and it has been exploited in the experimental set-up.

Forced Oscillation Technique (FOT) is a simple and minimally invasive method, useful to study the mechanical properties of the respiratory system. This observation is performed by measuring respiratory system's response to a forcing signal applied externally. The mechanical response is generally defined as impedance, as said before. To best measure Z, not Pressure's sources different from the externally applied, have to be recorded. To do this, the signal's frequency needs to be higher than the physiologic one ($f > 4Hz$). This so called "high-frequency FOT" permits the patient to breathe spontaneously and so it can be performed both in healthy and ill subjects.

FOT superimposes small pressure oscillation (about 1-5 cmH_2O) at the respiratory system while recording pressure and flow signals. This oscillatory pressure induces an oscillatory flow whose amplitude and phase is related to the mechanical impedance of the patient's respiratory system. As the respiratory system is not linear, the use of small oscillations permit to focus on a work point and to consider that as a linearization. Respiratory system impedance (Z_{RS}) is defined as the complex ratio between the applied pressure (P) and the resulting volumetric flow rate ($\dot{V}$) at the frequencies contained in the forcing signal. Therefore this complex function of frequency depicts the linear (or better, the linearized) behavior of the system in response to pressure oscillations and characterizes its mechanical response. Z_{RS} can be factorized in its real and imaginary part, respectively called respiratory Resistance and Reactance. R_{RS} represents the dissipative mechanical properties of the system, while the latter is related to the energy storage capacity and it is determined both by elastic and inertive properties. [Eq. 1. 13]

The technique was studied in '50s by Dubois [13] but only several years later it was greatly considered because of its technological difficulties to be applied. It gained interest because of patients who does not have to collaborate, and so it revealed a very useful technique for infants and very sick people unable to help doctors in measurement assessment.[14]

In recent years X_{RS} at 5Hz has been proved to be sensitive to alveolar recruitment and to permit the identification of the lowest PEEP that maintains lung volume recruitment minimizing lung mechanical stress in an animal model of ALI [15,16].

Moreover it has been shown that FOT is applicable in ventilated newborns and can provide useful information for tailoring the ventilator settings according to the patho-physiological characteristics of the patient. [17]

1.2 Newborns Peculiarities and Lung's Recruitment

1.2.1 Fetal and Postnatal Lung Development

Newborns' respiratory system is very different from adult's one. It offers a much greater physical difficult to breathe, i.e. it has a total respiratory impedance greater than adults' respiratory system. Thus, diseases that affect the small airways and cause large changes in peripheral resistance may be clinically silent in an adult, but can cause significant problems in infants (e.g., bronchiolitis). [19, 20] When birth occurs, some anatomical systems are not completely mature; the respiratory system is one of them. The development of the lung covers a period that begins with the appearance of lung bud in embryos and ends in the first infancy. [4, 20] The prenatal development of the lung can be divided in five periods described in synthesis in Figure 1. 14:

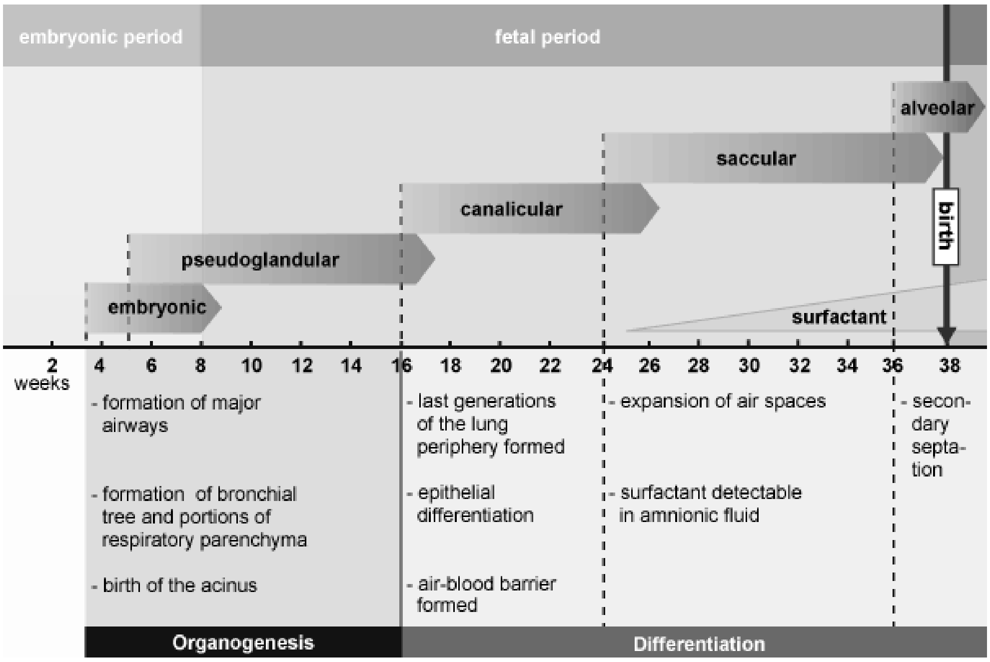


Figure 1. 14 Embryonic and fetal lung development

Most respiratory airways appear between the 16th week of gestation and birth while alveolar development starts relatively late in the 28th week of intrauterine life.

The development is not completed at 37th week of gestation, in fact the alveoli, as they exist in the adult, do not begin to appear until about 8 weeks of age. Most of them (85%)

are added after birth: when 2 months old, alveolar multiplication begins and continues to about 2–3 years of age [3]. Their number estimated at birth is about 3 to 5 times minor than in adulthood, when the number reaches 200-300 million. Notwithstanding the increase in alveolar number, the elastic fibers are still not mature; moreover, they do not reach adult values until adolescence.

Thus, even infants born in regular term have major difficulties than adults in normal breathing.

Birth does not correspond to a definite transition from one developmental stage to another, but it simply represents a critical event associated with dramatic changes in the lung function. A lot of factors influence the newborn survival in the post-natal environment: the fetal lung fluid which maintains the lungs in a distended state during fetal life needs to be absorbed, the lungs has to be filled with air, there has to be an adequate gas-exchanging surface area, the pulmonary blood flow has to greatly increase and the surfactant system has to ensure the lungs remain expanded by decreasing the alveolar surface tension [2]. Before the birth, the lung stands in a completely different status than the post-birth one. At birth instead, the fluid’s clearance from the airways and alveolar ducts happens through a mechanism of absorption forced by the increase in adrenaline and arginine concentration caused by the uterine contraction. The first breath is a gasp that generates a transpulmonary pressure of up to 80 cm of water and is the result of diaphragmatic descent. [4] This overcomes the surface forces at air-fluid interface and tissue resistance. Lungs expand and cause the traction of blood vessels from the interstitial lung tissue. It is the basis of the starting of reduction in pulmonary vascular resistance and of the redistribution of blood flows from the placental to pulmonary circulation. Moreover the first breath increases the alveolar oxygen tension, decreases the carbon dioxide tension and leads to the release of prostacyclum into the blood: vasodilatation follows. [4, 9] The fetal functional circulation starts to evolve and the ductus arteriosus constricts and permanently closes in the first day of life. Alveoli and blood vessels increase in number and size. Also the structure of the rib cage and the respiratory muscle undergo modification in the first periods of life to reach their complete development. The major portion of this growth however affects the respiratory zone. Later in life, the increase in lung volume is mainly due to expansion of alveoli, while lung size increases until growth of the chest wall is complete.

Central Control of Breathing

Considering breath-to-breath variability the breathing pattern of newborn infants is irregular, especially in those born prematurely. A considerable amount of maturation of control of breathing occurs in the last few weeks of gestation; this explains the high prevalence of apnea in infants born prematurely. Infants behave differently from adults in critical situations. For instance when suffering from hypoxia, the fetus tends to decrease his activity including breathing movements during intrauterine life. On the other hand adults respond to hypoxia with increased respiratory efforts. Even premature newborns are not able to sustain the physiological hyper-ventilatory response to hypoxemia for a prolonged period. [21, 22]

Upper and Lower Airways

In the newborn the epiglottis is relatively large, floppy, and high positioned in the pharynx so that it is in contact with the soft palate, thereby favoring nasal over mouth breathing. In addition, the relatively large tongue occupies most of the oropharynx which makes oral respiration even more difficult. Anyway, mouth breathing can occur if the nasal passages are obstructed. The configuration of the upper airways changes with growth. Any obstruction of the upper airway could cause significant dynamic inspiratory collapse of upper airway structure. Instead lower airway obstruction brings to forced expiratory effort, which causes an increased intra-thoracic pressure; this escalation has the result in limiting expiratory flow. [2]

Chest Wall

Because of its level of development referred to shape and deformability, the rib cage has a very limited level of contribution in newborns and infants. The sternum is soft too, providing an unstable base for the ribs. The highly compliant ribs are horizontally placed, and the intercostals muscles are poorly developed, so that the bucket-handle motion upon which thoracic respiration depends is impossible. Thus, the potential to increase the thoracic cross-sectional area is limited. The intercostal muscles and the diaphragm are antagonists at the costal margin, and the balance of control over the costal margin depends on the arch of the diaphragm. The loss in relative tidal volume compared to that of adults, is made up by greater expenditure in the work of breathing, even under normal conditions. The newborn and infant have high chest wall compliance, at least 5 times higher than that of the lung, contrary to the adult. [4] Normally, however, infants are constantly re-establishing lung volumes with crying and movement. In addition, the infant actively elevates the end-expiratory lung volume above the elastic equilibrium volume by expiratory braking.

Respiratory Muscles

The diaphragm is the most important breathing muscle, and any disease or physiological process which impairs diaphragmatic function will lead the individual to respiratory failure. The proportion of respiration performed by the diaphragm increases dramatically in REM sleep, because then the phasic and tonic inhibition of intercostal muscles causes the rib cage to move in an uncoordinated way and out of phase with the diaphragm. This is why even in adults, lots of respiratory problems occur during the rest time. The work of breathing in REM sleep is therefore high but this lasting effort is not compatible with infants' respiratory muscles, whose fibers are typically "fast twitch fiber". Maturational changes in the respiratory musculature occur, with increased mass and a progressive increase in the fatigue-resistant type I muscle fibers. [3] These differences between newborns and adults, coupled with newborns' higher energy requirement for growth, make easy to understand the reason of higher percentage of respiratory problems in infants.

1.2.2 Preterm Birth

The birth that occurs before the 37th completed week of gestation is defined as "preterm". Preterm birth comprises however a very heterogeneous group, with a wide range of gestational age. Common subdivision deals with "severe prematurity" and of "extreme prematurity", under 32 and 27 weeks respectively. Birth under 22 weeks is generally incompatible with life [23]. The preterm newborns are compounded by an incomplete structural development, a lower capacity to re-absorption of fetal lung liquid, the lack of surfactant and increased thickness of the alveolar-capillary membrane that interferes with gas exchange. Consequences of surfactant absence are several: increased surface tension, decreased compliance, presence of atelectatic areas and alveoli filled with transudated liquid. [9] All these are pathophysiological features typical in preterm infant with Respiratory Distress Syndrome (RDS). [24] With extreme preterm birth, the vascularization of the mesenchyme has just begun and there is still little differentiation of alveolar epithelium cells of type I and type II. This results in a minimal pulmonary functionality. Also if the birth is between 27 and 37 week of gestation, the lung is not already correctly developed and this results in the presence of an air-blood barrier that has just begun to dwindle, a vascularization still incomplete and immature type II cells. The birth at this level of development results in the presence of reduced number of alveoli, bronchial smooth muscle hyperplasia, abnormal capillary morphology and variable interstitial cellularity and fibro proliferative level. None of these stages of development is compatible with an autonomic ventilation and in most cases requires some form of ventilatory support [25]. Recent studies are said to investigate new methods to help newborns' breathing.

1.2.3 Recruitment Maneuvers

In the clinical management of ventilated infants it is very important to optimize lung volume applying sufficient pressure to keep the lung fully recruited without neither increasing the stress applied to the tissue that remains closed, nor over-distending alveoli through the tidal cycle. Recruitment is a strategy aimed at expanding, or re-expanding, collapsed lung tissue increasing the pressure applied to the lung, and decreasing it still maintaining enough pressure, to prevent subsequent 'de-recruitment'.

Because of the high variability of clinical conditions of preterm or pathological infants, the definition of optimal ventilator settings for all the newborns is impossible.

In order to recruit collapsed lung tissue, sufficient pressure must be imposed to exceed the critical opening pressure of the affected lung - in some areas of the lung, the pressures required may exceed 50 cm H_2O.

A strategy is needed to limit trans-alveolar pressures in the upper lobes, and provide sustained high pressures in the lower areas of the lungs sufficient to cause recruitment of collapsed tissue. Various ventilator modes - such as inverse ratio pressure-controlled ventilation, airway-pressure release ventilation, and even high-frequency oscillatory ventilation - have been used to promote recruitment, but a new strategy which prevent lung injury is required.

In order to optimize lung volume it is essential to open atelectatic alveoli and rapidly find the maintenance value of ventilator parameters that will keep them open.

Because each lung has peculiar characteristics and history, the definition of the correct therapy strategy must begin from the observation of the anatomical and physiological condition of the respiratory system of the considered infant. For example, currently the tidal volume is set according the infant weigh and arterial gas concentrations.

As written in par1.1.4 about the static P-V curve, the steeper the slope, the greater the compliance and then the distensibility of the lung. An important principle of lung recruitment is that the pressures required to open atelectatic alveoli are considerably higher than those required to keep them from closing again. So periodically recruitment maneuvers may keep lungs open without repeated application of potentially damaging forces during tidal ventilation [26]. Recruitment depends not only on the magnitude of transpulmonary pressure but also on the duration of its application and because of visco-elastance and other time dependent force distribution phenomena the full volume change is not realized for multiple tidal cycles afterward.

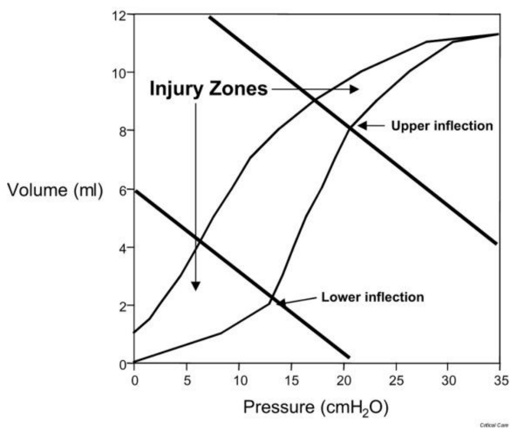


Figure 1. 15 P V curve where injury and safe region of recruitment are explicit

The optimal strategy to recruit lung has not be already defined. Once recruited, a correct ventilator strategy is necessary to avoid lung volume de-recruitment. Currently, pressure setting is based primarily on arterial oxygenation, inflation of the lung on the patient's chest radiograph and subjective assessment of chest wall movement.

Alveolar re-expansion can be achieved either by brief increases in mean pressure, termed sustained inflations (SI) or by the use of progressive PEEP for brief periods. However the techniques proposed in previous studies for this method of PEEP titration present a lot of limits. Oxygenation lacks specificity and is insensitive for detecting lung over-distension and tidal recruitment [25] while dynamic compliance will tend to underestimate true open lung PEEP, especially in the heterogeneous Respiratory Disease Syndrome (RDS) lung and it is strongly affected by both non-linearities and within-breath changes in lung mechanics and its value vary greatly with V_T.

However lung injured in different ways respond differently to recruitment maneuvers and not all the lungs are recruitable.

R_{RS} and X_{RS} measured by Frequency Oscillation Technique are very useful in optimizing the ventilator settings. A recruited lung could be less prone to ventilator-induced lung injury due to a higher normally aerated lung whose bulk of ventilation could be diverted and a lower heterogeneity, which is associated with a decrease in shear forces between healthy and injured zones.

Recruitment maneuvers will become part of the ventilator strategy when a method able to provide information about the timing of the maneuver is available.

The optimal recruiting values may result from a compromise between maximal lung recruitment and minimal over-distension (make reference to Figure 1. 15).

1.2.4 Sustained Lung Inflation (SLI)

Newborns' lung is full of fetal liquid, especially when the birth occurs preterm. The very first breath requires thus higher values of pressure, to overcome the high resistance and compliance. Preterm newborns have a respiratory system which is not completely developed and the lack of surfactant increases the respiratory effort. To assist them, ventilation is required but it is very important to avoid lung injury because of wrong ventilator settings.

To reduce VILI (Ventilator Induced Lung Injury) whatever is the ventilation support chosen for an infant, it is fundamental to recruit all the alveoli at birth. In the very last years, several studies have been done to evaluate the efficacy of the Sustained Lung Inflation (SLI) just after birth. This technique concerns the application of high pressure's peaks for few seconds, in order to permit the newborns to have a FRC (Functional Residual Capacity) sufficient to perform gas exchanges, through a mask or a nose-pharingeal tube. Common protocol uses the sustained inflation for 15 seconds at a pressure of about 25 cm of water, while during the rest period, it is only applied a PEEP of 5 cm H2O. Notwithstanding some results express no significant differences between SLI and the others VM, about lung injuries [25], SLI has been proved to be more effective than Intermittent Mandatory Ventilation (IMV) for newborns born in term, according to the FRC with asphyxia [27, 28] - make reference to paragraph 1.3.5 for the most common modes of ventilation. Moreover, recent studies tested the good scores achieved even in pre-term newborns. [25] A sustained inflation followed by early nasal continuous positive air-way pressure, delivered through a nasopharyngeal tube, is a more efficient strategy than repeated manual inflations with a self-inflating bag and mask followed by nasal continuous positive airway pressure on admission to the NICU. [29] The mentioned author performed a research with 25-32 weeks pre-newborns. His SLI's protocol resulted better than manual ventilation's one because at 72 hours after birth the need of mechanical ventilation was highly reduced. Linder et al. instead tested 25-28 weeks pre-borns a SLI protocol in NICU with 20-30 cmH_2O for 15s and the results were not different from those gained with IM Ventilation. Furthers animal tests are rich of interest and testify a synergic effect using both PEEP and SLI to reach and maintain good level of FRC. [30, 31] Thus research needs more efforts and more experiments to find a solution to reduce Respiratory Distress and to evaluate the clinical utility of the adopted technique.

The possibility to monitor what happens in lungs during the application of these recruiting techniques, should be very useful in order to define an optimal procedure to maximize the final FRC and to minimize the stress applied to pulmonary tissue. Forced Oscillation Technique could be useful to measure newborns' mechanical properties because is minimally invasive and does not require direct collaboration of patient. [32, 33] Moreover, in this way it is always possible to monitor the respiratory system while recruiting and so it is possible to choose run time the best ventilation mode and ventilation parameters.

1.3 Mechanical Ventilators

1.3.1 The Neonatal Intensive Care Unit (NICU)

Figure 1. 16 Newborn intensive care unit

An Intensive Care Unit (ICU), whose synonyms are also Critical Care Unit or Intensive Treatment Unit (ITU), is a special department of a hospital that provides intensive care medicine. They are dedicated to patients with the most serious injuries and illnesses. These patients need constant, close monitoring and support from specialist equipment and medication in order to maintain normal bodily functions.

Figure 1. 17 ICU Central control area

The specialized intensive care unit deals with newborns called NICU.

When a baby enters the world, not only respiratory system change dramatically from the way it functioned during fetal life, but also:

- The cardiac and pulmonary circulation changes.
- The digestive system must begin to process food and excrete waste.
- The kidneys must begin working to balance fluids and chemicals in the body and excrete waste.
- The liver and immunologic systems must begin functioning independently.

The NICU combines advanced technology and trained healthcare professionals to provide specialized care for the tiniest patients. NICUs may also have intermediate or continuing care areas for babies who are not as sick but do need specialized nursing care. Ten to 15 percent of all newborn babies require care in a NICU, and they are admitted within the first 24 hours after birth. In Italy, nearly 12 percent of babies are born preterm, and many of these babies also have low birth weights (in USA about 13%).
By the 1970s NICUs were an established part of hospitals in the developed world. Anyway, only after 10 years more NICU gained importance for the most of people. Before 1980s in fact the number of infants born in their houses was significant. On the other hand, NICU environment at the very beginning was very hard to support both for parents and sons. A 1979 study showed that 20% of babies in NICUs for up to a week were never visited by either parent. Nowadays NICUs concentrate on treating very small, premature, or congenitally ill babies. Neonatology and NICUs have greatly increased the survival of very low birth weight and extremely premature infants. In the era before NICUs, infants of birth weight less than 1400 grams (usually about 30 weeks gestation) rarely survived. [31] Today, infants of 500 grams at 26 weeks have a fair chance of survival.

Although each country manages NICUs in different levels of care, it is easy to summary all the guidelines into three stages:

- Level I basic
- Level II specialty
- Level III: advanced specialty

The first level does not deal with less dangerous situations, but instead it deals with more developed newborns. In fact nursery has the usual capabilities to provide neonatal resuscitation at every delivery, evaluate and provide postnatal care to healthy newborn infants, born at 35 to 37 weeks' gestation and who remain physiologically stable stabilize newborn infants who are ill and those born at 35 weeks' gestation until transfer to a facility that can provide the appropriate level of neonatal care.
Level II special care nursery unit has the capabilities to resuscitate and stabilize preterm and/or ill infants before transfer to a facility at which newborn intensive care is provided. It

provide care for infants born at 32 weeks' gestation and weighing 1500 g who have physiologic immaturity such as apnea of prematurity, inability to maintain body temperature, or inability to take oral feedings or who are moderately ill with problems that are anticipated to resolve rapidly and are not anticipated to need subspecialty services on an urgent basis. Level II can provide mechanical ventilation for brief durations (less than 24 hours) or simply continuous positive airway pressure to help the little patient to breathe. Level III is the more critical level of care. It provides comprehensive care for infants born at 28 weeks' gestation and weighing 1000g. Mechanical ventilation is very important to let doctors perform minor surgical procedures such as placement of central venous catheter or inguinal hernia repair. Also advanced imaging, including computed tomography, magnetic resonance imaging, and echocardiography is usually performed at this level of attention. Complex cardiopulmonary bypass can be done.

Fundamental device in every NICU is mechanical ventilators and respiratory supports. There are very different kinds of ventilators and different modes of ventilation.

1.3.2 Classification of Ventilators

Ventilators have been used in patient management since about the '50s. Since the born of these devices, different classifications have been made. [34, 35, 36] Ventilators could be grouped by the role they act during ventilation, the power source or by the different breathing patterns delivered to patients, the king of feedback control. Ventilator is basically a box powered by an energy source which provide breath gas to the user. The operators can set the parameters on the control panel. These settings detetmine the kind and the amount of gas the patient receives.

Assister Ventilators and Controller Ventilators

Respiratory mechanical devices can be grouped in two cluster because of their role: *controller* or *assister*.

When a patient is connected to an assister ventilator, his or her respiratory ventilation is independent to the functioning of the ventilator, as he can breathes alone. However, at the same time the respirator detects the breath and assists the attempt. Eventually, ventilator performs a minimum breath frequency and executes a breath whether the patient does not perform a mechanical act in the check time. In other words, if the backup frequency is set to 10 bpm (breaths per minute), the maximum lag between an interval and another will be 6 seconds – make reference to Figure 1. 18 . If the patient does not breathe in 6 seconds, the ventilator starts to directly control the mechanical act.

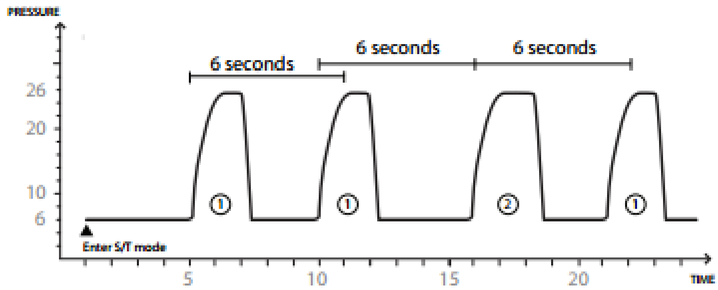


Figure 1. 18 Spontaneous and Temporized mechanical ventilation.

Controller ventilators act as direct performers of breaths. The device sets the ventilatory cycle and any tendency toward spontaneous ventilation on patient's part, does not affect the machine functioning.

Pneumatically Powered Ventilators or Electrically Powered Ventilators

The Power source provide the machine with the energy to perform the work required to ventilate a patient. According to this discriminator, there are two categories: pneumatically powered ventilators or electrically powered ventilators. [35]

The pneumatically powered ventilators connect to high-pressure gas sources and use this pressure to power gas flow to patient. In general, pneumatically powered ventilators used in an intensive care unit (ICU) use 50-psi gas sources (generally 35 to 100 psi) and have built-in reducing valves so that the operating pressure is lower than the source pressure. With pneumatically powered ventilators, gas flows down a pressure gradient from the wall, through the pressure-reducing valves, and to the patient without the need of a mechanical device such as a piston. Basically ventilators used in ICUs are pneumatically powered.

The elettrically powered ventilators most often use standard electrical outlets. They also may have internal direct current (DC) batteries, which can provide electrical power during patient transport or in the case of a power failure. This group use electricity to power internal motors to provide ventilation. Air is not compressed but moved, usually, by turbines. These devices work alone without any air supply, and they need foam filter in order to prevent eccessive pollen to reach patient's airways.

Negative Pressure Ventilators and Positive Pressure Ventilators

Negative pressure ventilators provide the patient with a negaive pressure. Negative pressure is more physiological; the thoracic area and create a sub-atmospheric pressure around the chest wall. This negative pressure is transferred to the space within the thorax, producing a pressure gradient along the trachea that results in air entering the lungs. Notwithstanding this, they are by necessity large and limit access to the patient for theraphy.

Nowaday NPVs are thus rarely used in the ventilation. The pressure at the mouth is the atmospheric and this gradient of pressure between mouth and alveoli let air flows into the lungs. To permit exhalation, the pressure surrounding the chest is let to return normal and the physiological way of expiration is so enabled. These ventilators currently are not widely used, but they occasionally are used in the home as an alternative form of ventilation for ventilator-dependent patients. Examples of negative ventilation are the iron lung and the chest cuirass (Figure 1. 19).

Figure 1. 19 Chest cuirass

Positive pressure ventilation is by far one of the most common mode of ventilation today. A pressure gradient must exist for gas flow to occur. In order to simulate the contraction and the descent of diaphragm at the beginning of the inspiration, a super atmospheric pressure is created at the mouth, and the intra-alveolar pressure is therefore forced to ambient at the beginning of the breath. As a result, air flows into the lungs, expanding them and the chest-wall. Expiration instead occurs as in physiologic way, such as because of passive relaxation of the respiratory muscles. [37]

Open- and Closed-loop Ventilators

Mechanical, pneumatic, and electronic devices in the ventilator constitute the control unit. These systems can create an open loop or a closed loop system.

The terms above mentioned stand for the level of control within a ventilator. Ventilators uncapable to autoset parameters not to maintain fixed values of pressor and flow, are called open-loop systems. When the operator establishes a setting, such as the tidal volume, the unit delivers the amount of volume. Actually, it might never reach the patient. An open-loop system is not able to discern whether the amount set is delivered or not and therefore cannot respond to the difference. Closed-loop system are, on the contrary, intelligent systems which use a microprocessor to control function. For example, the system control both the quantity of gas set and the one delivered and after a comparison, behaves in order to nullify the difference. It is closed because it compares the input to the output and "closes the loop". These systems are also called feedback systems or servo-controlled systems. [38] A direct example of closed-loop control is the Auto-set point. With auto set point, the output of the ventilator is forced to follow or match an operator preset input. For example, inspiration starts, out delivering a constant pressure. This action is controlled by a series of "if-then" statements. If the set volume has not been met on time and flow decreases lower than the pre-set value, the ventilator switches to volume ventilation and maintains the flow at the set flow value.

1.3.3 Physical Characteristics

Sensors

To best control patient breath, several sensors are implemented in ventilators. It is easy to recognize oxygen sensors, CO_2 sensors, pressure and flowmeter sensors. Brief explanation will be given about the first sensors mentioned, while a wider one is reserved to flowmeters.

Oxygen and CO_2 sensors

To understand whether patient is regularly breathing, it is mandatory to check the blender FiO_2. It is the Fractional Oxygen percentage to deliver to patient. if he or she only lack mechanical respiration, Oxygen percentage wil be 21%, such as in atmosphere. On the contrary, if the damage is even chemically and the patient is not able to provide sufficient oxygenation with the common O_2 percentage, it is necessary to enrich oxygen percentage and thus to augment FiO_2.

At the same time, a sensor to detect how much CO_2 has been expired is very useful. If the ventilator uses a 2-breathing circuits, CO_2 is sensed in the expiratory tubes. To detect CO_2

is very important and useful. Carbon dioxide chemically binds to hemoglobin -forming carbaminohemoglobin. This implies two consequences. Firstly, a decrease in hemoglobin's affinity for oxygen. However, the main consequent is an increment in PH-acidity which is very dangerous.

Pressure Sensor

Pressure is very important to be sensed because the doctor need to be sure of the amount of cm of water provided to patients. Pressure sensor is an useful instrument used by the control circuit of ventilators, to deliver the exact amount of pressure. It is usually sensed in proximity of patient airways –make reference to Figure 1. 24 in the next pages.

Flowmeters

This category of devices measure directly the air flow. To fulfill several requirements, many kinds of flowmeters are commercially available, and they are based on different functioning principles. Flowmeters measurement should not be influenced by gas composition, humidity, viscosity and temperature. These parameters, in fact, vary between inspired air, which is at environmental conditions and doesn't contain carbon dioxide, and expired air, which is at body temperature, water vapor saturated, poorer in oxygen and contains carbon dioxide instead. Also, as said in the previous paragraphs, sometimes gases are added to the inspired mixture. In order to pass from the flow information to volumes, flowmeters integrate electronically the flow signal.

Pneumotachometers

Figure 1. 20 Pneumo-tachometer

This kind of measuring flow device exploits Poiseuille's Law, which states that, in cylindrical rigid tubes, the laminar flow of a Newtonian fluid gives rise to a pressure drop proportional to the flow. As said, Poiseuille's resistance (R) is the ratio between pressure and flow, and it is determined by the length (l) and the radius (r) of the tube and by the viscosity (μ) of the fluid. These relation have been described in [Eq. 1. 5] and [Eq. 1. 6], which however are reported below:

$$\Delta P = R * \dot{V}; \quad R = \frac{8\mu l}{\pi r^4}$$

Pneumotachometers make the air flow to pass through a bundle of brass tubes of very small diameter, and measure the pressure drop between the edges of these tubes. If the air flow through this resistor keeps laminar, Poiseuille's law holds and the pressure signal we measure is proportional to the air flow. Whether the air composition and the geometrical features of the tubes are known, Poiseuille's resistance is known too. The flow being laminar is very important in pneumotachometers; this is the reason why the mechanical resistor is obtained with tubes of very small diameter, according with the Reinolds number. Also the morphological design is dictated by the tradeoff between the need of minimizing turbulence and to keep small the dead space of the device. In fact, flow remains laminar as far as fluid velocity is low. Being forced expiratory maneuvers performed at considerably high flow rate, the low flow velocity would be warranted by a large cross sectional area. On the contrary, however, the sectional area cannot be large because of the need to fit patient's mouth; moreover, in the tract between patient's mouth and the resistor the cross sectional area increases yet. The tradeoff between dead space and turbulence dictates exactly the angle this tract slopes up. Small angles do not determine turbulence but require a greater length to reach the desired sectional area. The effect should be the dead space of the device to be large… it is thus necessary to strike a balance. Usually the small tubes representing the resistor are warmed up at body temperature (37º C) to prevent water vapor in the expiratory phase from condensing inside the tubes, which would bring an unpredictable increase of the resistance. Nowadays some manufacturers have replaced the metal tubes with ceramic resistance elements, because ceramic provides better heat conduction to the gas (in order to have measurements at body condition) and because, being ceramic porous, water tends to be adsorbed.

In continuous ventilation monitoring, whatever resistor is placed in the air flow expired from the patient. Devices would have very rarely a linear and non-turbulent flow in this case. In order to compensate this fact, the signal is electronically conditioned and non-linearity introduced by turbulent flow are corrected.

A part eventual flow turbulence, the fact that gas composition and temperature vary between inspired and expired air would represent another source of error, because these

factors influence the mechanic resistance. Luckily, when the patient inspires room air, the separate effects of temperature
and of gas composition tend to cancel out each other. However, to not influence the respiratory performance of the patient, the resistance to air flow opposed by pneumotachometers must be always small. This means that the pressure drop which is measured, being proportional to this resistance, is also very small, therefore the need of very sensitive differential pressure transducer.

Hot-wire Anemometers

Another typical device to measure airflow velocity is the hot-wire anemometer, observable in Figure 1. 21. When a fluid flows around an element with different temperature, there is a heat transfer from the heated element to the fluid, which is proportional to the flow rate. The measurement of this heat transfer can be used for determining the respiratory flow. The heated element, generally a thermistor, is kept at high temperature (100-400ºC) by a current passing through it. Current I passes in a very thin platinum wire (diameter dimensions 10^{-6}m) horizontally positioned in the field of flow, with a length of 1-2mm. This makes the device to respond promptly to changes in the flow rate and the bandwidth of the these devices extends from DC to several kilohertz. At the same time however, the small dimensions make the heated element extremely delicate, but mostly they imply that the air flow derives from the measurement of fluid velocity in a small portion of the cross sectional area.

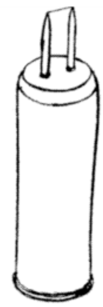


Figure 1. 21 Hot-wire anemometer

The greater the air flow, the more the element is cooled; dispersed power W is proportional to the second power of current I and to the resistance R_W of the wire:

$$W = I^2 R_W$$

[Eq. 1. 15]

R_w depends on:

$$R_W = R_{REF}[1 + \alpha(T_W - T_{REF})]$$

[Eq. 1. 16]

Which express the linear proportionality of wire resistance R_w from temperature; the offset is REF temperature. When airflow passes through the platinum or tungsten wire, heat tend to decrease because of thermal convection. This can be explained in a more technical sight by the equation:

$$W = \hbar * A_W * (T_W - T_F)$$

[Eq. 1. 17]

Combining [Eq. 1. 14], [Eq. 1. 15] and [Eq. 1. 16], it is easy to achieve the following:

$$I^2 R_W = \hbar * A_W * (T_W - T_F)$$

[Eq. 1. 18]

Where h is the convection constant, A_w the wire surface, and T_w and T_F stand for wire and flow temperature, respectively. Heat flow is proportional, among all, to h which depends on flow density, viscosity, specific heat too:

$$\hbar = a + b * v_f^2$$

[Eq. 1. 19]

a and *b* are parametric values, v_f is the flow velocity. Thus the effect is a decreasing in resistance. Actually there are two way of functioning:

-measure *R* changes, by keeping the *I* current constant;

-measure the exact amount of *I* to maintain the *R* constant.

In the first case flow velocity depends on initial current and flow temperature, while in the second flow velocity depends on both wire and flow temperature.

E.g. in the second case is therefore needed a more intense current to keep constant the temperature. The measurement of this current provides an estimation of the heat transfer, hence of the air flow. The heat transfer from the heated element to the fluid flow is influenced by several variables, including gas temperature, gas viscosity and thermal conductivity, which in turn depends on the composition of the gas mixture.

Combining [Eq. 1. 17] and [Eq. 1. 18] in [Eq. 1. 15], the second to last equation is created:

$$a + b * v_f^2 = \frac{I^2 R_{REF}[1 + \alpha(T_W - T_{REF})]}{A_W(T_W - T_f)}$$

[Eq. 1. 20]

Generally these devices are used only for measuring unidirectional flow (either inspiratory or expiratory air flow), because in their most simple implementation they are not able to discriminate flow direction, therefore temperature and viscosity differences, which exist between inspired and expired air, do not concern these devices. If the flow is laminar the heat transfer is more linear and predictable, however the relationship connecting current to air flow remains in any case not linear, hence the need of a linearizing circuit.

When instead the device is requested to sense flow direction, a second heated element is added inside the flow, as well as the necessary circuitry for discriminating the direction. This the case of general respiratory ventilators. In fact this is the required to measure a mono-dimensional, bi-directional gas flow. The second element in this case must not be cooled by the flow, but it varies its resistance only according to the baseline temperature. The two thermistors are connected in a Wheatstone bridge for comparison.
In this case the functioning principle is the temperature constant, the equation simplifies:

$$a + b * v_f^2 = \frac{I^2 R_{REF}[1 + \alpha(T_W - T_{REF})]}{A_W(T_W - T_f)} = g(T_W, T_f)$$

[Eq. 1. 21]

So flow velocity depends on wire and flow temperature.

Dynamically, in turbulence measurements it is generally assumed that the frequency response of the hot-wire system is similar to a low pass filter with constant response at low frequencies and a graduate roll off near the cut-off frequency. The hot-wire system is calibrated using either a ‘static’ calibration method or a “dynamic” calibration method [39] but at low frequencies around 1 Hz. The output voltage is then related to the mean velocity usually by a polynomial of 3rd to 5th order depending on the calibrated velocity range. [40, 39]

Pneumatic Circuit

It consist of a series of hoses which direct gas flow both within the ventilator and from the ventilator to the patient.

Internal and Control Circuit

The internal circuit conducts gas generated by the typical power source, passes it through various mechanical or pneumatic mechanism, and finally directs it to the external circuit. In a single-circuit ventilator, the gas enters the ventilator and goes directly to the patient. The mechanism is simple and can be easily evinced from the Figure 1. 22:

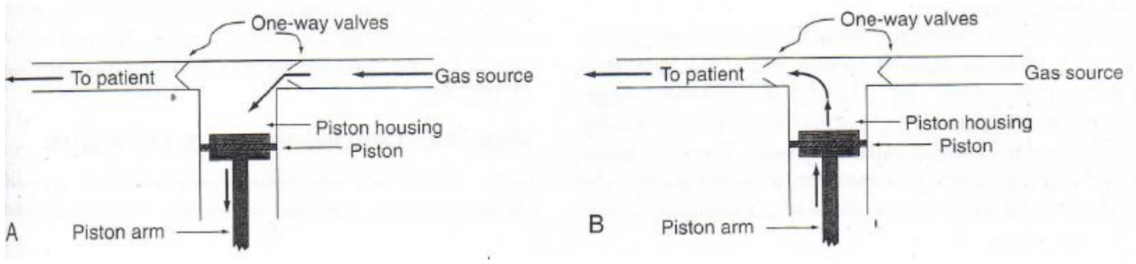


Figure 1. 22 Control circuit

one way valves let air flows from gas source, when the piston arm descends and creates an appropriate gradient of pressure. Thus, the other one-way valve let air flow toward the patient when the piston arm moves upward into the cylinder. Double-circuit ventilator, instead, consist of two gas sources. The former goes to the patient from a bellows or a bag, the latter actively compresses this bag. To be considered Figure 1. 23:

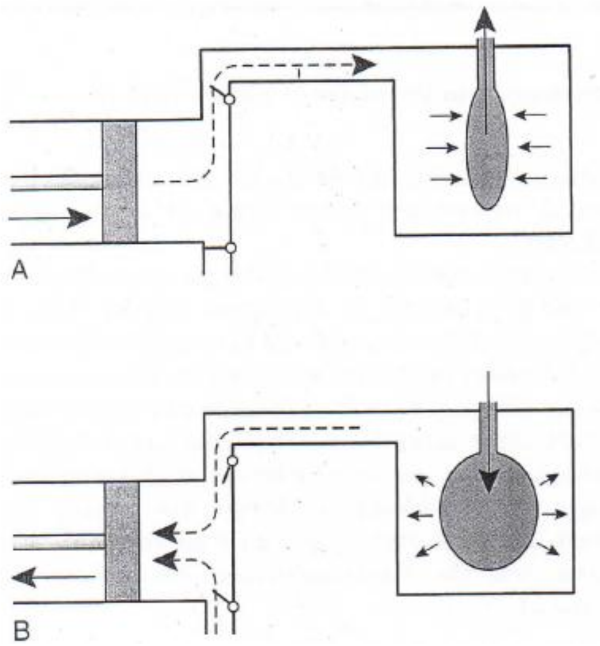


Figure 1. 23 Simple diagram of a piston driven: Fig. A forward stroke which causes the gas to flow to the chamber containing the bag. Fig.B backward stroke with gas entering the piston cylinder.

To use the drive mechanism and the gas delivery valves previously described, the ventilator must be programmed to manipulate pressure, volume, flow, and time to deliver a breath (inspiration) to the patient. A control circuit performs this function. It is part of the internal circuit and it consists of mechanical devices such a spring-loaded valves and of pneumatic systems such as pressure regulators or fluidic components. Current ICU ventilators tend to use sophisticated control circuits composed of programmed microprocessors and flow valves controlled by these microprocessors.

External Circuit

The external circuit conducts the gas from the ventilator to the patient and from the patient through an expiratory valve to the room. The external circuit is usually called ventilator circuit or patient circuit. Gas flows both through the main inspiratory line to the patient and to the expiratory line, while during exhalation, airflow exits from the exhalation valve. The valve is closed during the inhalation phase by the air pressure which insufflates a balloon that occludes the air gap. There are different kinds of circuits, according to the different modes of ventilation. For the most ICU ventiators, there are used external 2 circuit tubes: one for inspiration and one for expiration, such as in Figure 1. 24. It can be seen the two circuit for inhalation and exhalation, their connection (Y) and an HME filter to augment the moisture.

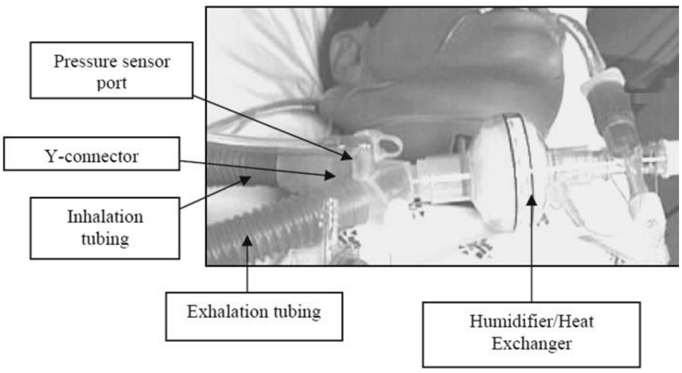


Figure 1. 24 Ventilation circuit

Breath generator

Pistons

For ventilators, there have been used two design of piston: direct drive and indirect drive. The former design consist in special gearing connects an electrical motor to a piston rod or arm. The rod moves the piston linearly, or circularly. With respect to the former mode, the resulting waveforms can be simply described as a rectangular flow wave, ascending ramp volume wave and simil-linear ascending pressure waveform. This kind of driving is typical of high-frequency ventilators. The latter instead, generates an half period sinusoid flow waveform and a not linear rising pressure waveform.

Spring-Loaded Bellows
Another volume design uses a spring-loaded bellows as the force that delivers the breath. The operator can tighten the tension of the spring to increase the force exerted against bellows and the pressure delivered to the patient. When the working pressure is low, especially when the system resistance is high, the flow curve descends during inspiration (at maximum delivery pressure, flow is 0).

Control Valves

Flow-Control Valves
Most current ICU ventilators use valves that precisely control flow to the patient. High Pressure source of air and oxygen are mixed and delivered to accumulator chambers, places similar to gas cylinders. They hold about 2 or 3 L of gas. Therefore, gas is sent through one or more valves that control the inspiratory gas flow. Flow-controlling technologies can be divided into three groups: proportional solenoid valves, stepper motors with valves, and digital valves with on/off configurations.

Proportional Solenoid Valves
Proportional Solenoid Valves control flow through an on/off switch. A proportional solenoid valve operates on a basic principle of physics concerning electricity and magnetism.
Whenever a current flows through a wire, a magnetic field would have been created. Adding an iron rod attached to the coiled wire, increase further the magnetism and produces an electromagnet.

Stepper Motors with Valves
Stepper motors can move quickly discrete steps in order to open or close a valve. The lever arm is connected to a rotating metal wheel, which permits movement in fixed steps, thanks to its gears. As the wheel moves, the arm moves forward and back, opening and closing a plastic tube and thus controlling the flow of gas.

Digital Valve On/Off Configuration

With a digital valve on/off configuration, several valves operate simultaneously, assuming either an open or a closed position. Each valve produces a specific flow by opening or closing an orifice of a particular size. Depending on which valves are open, the amount of flow can be varied.

In addition to the primary drive mechanism of a ventilator, other components are important to its operation. Such a component is the expiratory valve.

Peep Valve

Expiratory valve are normally closed during inspiration, directing gas flow into the patient's lungs, and open during exhalation, allowing the patient to exhale through a valve. Optimally designed expiratory valves allow unrestricted flow from the patient. This component permits the application of positive pressure during the expiratory phase, in order to avoid the lung to collide. This technique is referred to positive end-expiratory pressure, or PEEP. Use of PEEP elevates the baseline pressure above ambient (zero). Baseline pressure is the pressure sustained during expiration. The PEEP valve generally is located inside the ventilator housing. They can be magnetic and electronìmagnetic, or diaphragmatic.

Diaphragm expiratory valves commonly incorporate a large-diameter diaphragm to control expiratory gas flow and PEEP. During inspiration, the ventilator pressurizes the diaphragm and closes an expiratory orifice. During normal expiration, the pressure against the diaphragm (balloon) is released, and patient's expiratory gas flow naturally. [41] When PEEP is applied, a proportional pressure is held against the diaphragm so that the pressure at the end of exhalation is equal to the PEEP value set by the operator. [42]

Some PEEP valves use magnetic or electromagnetic forces to oppose gas pressure. The difference between the two principles is very slight. For example, electromagnetic valves have solenoids in which the amount of electric current that flows into them is regulated by a rheostat. The solenoid creates a downward force through an actuating stem. The actuating stem pushes against a diaphragm, which opposes expiratory gas flow.

1.3.4 Components of Breath Delivery

Briefly explained the main functioning princiles of ventilators, it is possible to discuss about the kinds of breath delivery and the parameters of ventilation directly controlled.

Type of Breath Delivery (Breath Classification)

Breathing can be provided by the ventilator and by the patient and therefore there are many kinds of breath delivery. If the ventilator controls the breath, the breath is called a mandatory breath; in other words, the ventilator can start and end the breath, or it can simply end the breath. An assisted breath, on the other hand, is one in which all or part of inspiration (or expiration) is performed by the ventilator doing work on the patient. Spontaneous breath is controlled by the patient who manages the transition from inspiration to exhalation.

Control Variable

The main parameter or variable controlled by the ventilator - for any given breath - is considered the control variable**.** This is the parameter the ventilator adjusts or manipulate to deliver inspiration. Pressure, volume, and flow are the three main control variables. Most current ICU ventilators function as flow controllers. These systems are very precise and can measure and adjust gas flow hundreds of times per second. In the clinical setting, the two control variables most commonly used are volume and pressure. Operators select either a volume they wish to deliver or a pressure they wish to deliver. Selection of a volume is called *volume targeted ventilation,* which is also known as *volume ventilation.* Selection of a pressure: is called *pressure-targeted ventilation,* which is also known as *pressure ventilation.*

Control variables manage the phases of breath. Ventilatory breath can be divided into four phases:

- the trigger variable begins inspiration;
- breath delivery, or inspiration, accomplished by the settings of parameters in the ventilator’s front panel (control in pressure, volume or flow) ;
- the cycle variable ends the inspiratory phase and begins exhalation;
- the limit variable limits the value for pressure, flow, volume or time during delivery of inspiration.

Pressures and flows can be controlled during exhalation; therefore the ventilator also can be involved in the expiratory phase.

Beginning of Inspiration: the Triggering Variables

The trigger variable begins the inspiratory phase, as briefly said. Ventilator breaths can be time, pressure, flow, or volume triggered. For istance, if the trigger variable is decided to be the time, the beginning of inspiration will be at a certain period. The triggering takes note of patient triggering too, such as to the starting of atient to inspirate. In this case, the device can be adjusted to sense the patient's inspiratory effort. Anyway, pressure and flow are the most common variables used for patient triggering, but volume and neural triggering also can be used. The former occurs when the ventilator senses a stop in pressure below baseline in the circuit. The latter instead happens when a decrease in flow is detected. In some ventilators a pneumotachograph is located between the ventilator circuit and the patient to measure flow.

Inspiratory Phase

One of the most important ventilator functions is delivery of inspiratory gas flow. With **pressure ventilation (PV),** pressure is selected as the control variable. [43] The pressure remains Constant during inspiration, but the volume and flow delivered may change if the patient's lung characteristics change.

Termination of Inspiration Phase: Cycling Phase

The phase variable measured and used to end inspiration, is called the *cycle variable*. A breath can be pressure cycled, time cycled, volume cycled, or flow cycled.

Pressure Cycling

A breath is considered pressure cycled when inspiratory flow ends and thus the expiratory phase starts once a pre-decided pressure is reached. It is not a common method used to cycle an ICU ventilator. When instead time is the phase variable used to end inspiration and allow expiratory gas flow to occur, the breath is said to be *time cycled*. Many current ICU ventilators are time cycled; the user sets a control called *inspiratory time*, which commonly determines the length of inspiration. If the breath is taking too long to delivery, the ventilator ends the breath based on a fixed time.

Volume Cycling

During volume ventilation, flow is delivered from the ventilator to the patient circuit and the patient until a specific volume has been delivered from the ventilator. As soon as the volume has been delivered, inspiratory flow ends and exhalation begins. Even though the ventilator may measure a specific volume output during volume-cycled ventilation, the amount delivered to the patient may be less. This discrepancycan be due to leaks in the system or compression of some of the volume in the patient circuit (tubing compressibility. Volume can also be lost as a result of a phenomenon called tubing compressibility (tubing compliance factor). Part of the tidal volume delivered from the ventilator contributes to the expansion of the circuit. This amount of volume never reaches the patient. This "lost" volume is important for infants and small children. For this reason, infant patient’s circuits are much smaller in diameter and are made off a lower compliance plastic and thus expand less. Most current ICU ventilators can automatically measure and calculate the loss of volume from tubing compressibility and compensate by increasing actual volume delivery.

Flow Cycling

A further criterion for terminating inspiration is flow cycling. Pressure-support ventilation (PSV) is the most common mode that uses flow cycling. Historically, PSV was the only mode that was low cycled out of inspiration. Now some ICU ventilators allow pressure-limited, time-cycled breaths.

Inspiratory Pause

The inspiratory time can be extended or delayed by keeping the expiratory valve closed for a while, in order to exclude gas flow to leave the circuit. In infant ventilators the pause is usually performed by a pressure-relief valve. The pressure relief valve opens during the inspiratory phase and allows the pressure in the circuit to be mantained at a constant level. Excessive pressure is let into the room.

Normally, when inspiratory flow stops during ventilation, the expiratory valve opens, allowing expiratory flow to begin. It could be triggered by a pre-fixed baseline pressure. Baseline pressures usually are positive end-expiratory pressure (PEEP) or continuous positive air way pressure (CPAP).

1.3.5 Basic Modes of Ventilation

Breath can be delivered to a patient in several ways, which not only describes the waveform of stimulation but also how the breath is controlled, cycled and which kind of parameter triggers. The *mode of ventilation* represents the interaction between the ventilator and the patient. It describes three essential components:

- The breathing sequence and the control variables in a breath
- The control type used between breaths
- The control variables.

One classification is the one formuled by Chatburn [35, 36] but because of the name of patterns given by the different factories, it is very difficult to univocally assign a name to the kind of breath. Anyway, he established some guidelines to avoid or minimize the risk of misunderstandings.

The main breath-control variables are volume, pressure and dual control. In volume control, volume remains constant but pressure can vary. On the other hand, in pressure control it remains constant and volume can vary. The dual control, instead allows the ventilator to switch from pressure control to volume control or vice versa within a breath. [43, 44] Very advanced control types use complex instructions, which are referred to as computational logic. It is a description of the relationship between the settings selected by the operator, the monitored information feed-back to the ventilator computer, and the final output or breathing pattern delivered.

Adaptive set-point control ventilation is an useful skill very often implemented in respiratory devices. They can switch modes based on monitored information. When this feature is turned on, the ventilator can switch from volume control to volume support, thus allowing the patient more control over his breathing.

Continuous Positive Airway Pressure (CPAP)

Continuous positive airway pressure is very close to PEEP. CPAP provides positive airway pressure, but it is restricted by definition to use in spontaneously breathing patients. Some CPAP devices are designed for home care to treat sleep apnea, such as the one depicted in Figure 1. 25. CPAP also can be used for certain types of hospitalized patients who are able to breathe spontaneously but who need help with oxygenation. In these cases, CPAP generally is provided by a ventilator or CPAP device. The gas is usually warmed and humidified. A threshold resistor is attached to the expiratoty end of the system. Sometimes a safety pressure-release valve is incorporated into a CPAP system. The safety pressure-release valve setting is slightly higher than the desired CPAP level. For example, if the CPAP is 10 cm of water, the safety pressure release might be set at 15 cm of water. A variety of problems can occur with a CPAP system; therefore the operator must make sure the safety systems (i.e., pop-off and pop-in valves) are in place and the patient is monitored. An example of these problems is the inadequate flow to the patient because of obstruction of the expiratory threshold. CPAP is one of the easiest mode of ventilation. However, it has been created even an AUTO-CPAP mode. In a prefixed range of max-min CPAP, the ventilator sets an optimal value of continue airway pressure thanks to sensor feedback.

Figure 1. 25 Philips Respironic's (auto) CPAP ventilator

Bilevel Positive Airway Pressure(BiPAP)

Bilevel positive airway pressure is similar to CPAP in that both breath delivery techniques provide positive pressure during inspiration and expiration. With CPAP, the pressure tends to stay at a fairly constant baseline with slightly negative pressure as the patient breathes in and a slightly positive pressure as the patient breathes out. With Bilevel positive airway pressure, the inspiratory positive airway pressure (IPAP) is higher than the expiratory positive airway pressure (EPAP). The normal interface between the unit and the patient is a mask. In many cases the unit can determine the amount of leakage and increase the flow output to compensate. It can continue its normal triggering and cycling, even when a small leak is present.

Intermittent Mandatory Ventilation (IMV)

IMV is designed to deliver volume or pressure-targeted breaths at a set minimum frequency, so it is a time triggered mode of ventiation. Between mandatory breaths, the patient can breathe spontaneously from the ventilator circuit without getting the set volume or pressure from the ventilator itself. During this spontaneous breathing period, the patient breathes from the set base line pressure, which may be ambient pressure or a positive baseline pressure (usually PEEP or CPAP). Because patients have an opportunity to breathe spontaneously, IMV is commonly used for patients who can provide part of the ventilatory work.

A slightly different kind of ventilation, is the synchronized one (**SIMV**). With SIMV, mandatory breaths could be either patient or time triggered. In the case the patient is the trigger, the ventilator synchronously delivers the mandate or breath. On the contrary, a time-triggered mandatory breath is delivered.

Intermittent Positive Pressure Ventilation (IPPV)

IPPV requires compressed gas to be delivered under positive pressure into a person's airways until a preset pressure is reached. As in IMV case, the variant of this mode of ventilation is the synchronized one; with SIPPV, mandatory breaths could be either patient or time triggered.

Pressure-Support Ventilation

PSV,or PC-PSV, is a spontaneous breath type that allows the operator to select a pressure to support the patient's work of breathing. As in pressure-controlled continuous mandatory ventilation (PC-IMV), the ventilator delivers a high flow of gas to the patient when a breath is triggered. As the lungs fill, the flow and the pressure gradient between the machine and the patient decrease [45, 43] If a leak is present in the system, the ventilator increases the flow to maintain the set pressure. As a safety feature, ventilators have a back-up time cycle that usually is a set time between 1 and 5 seconds. A way to estimate the starting pressure level is to calculate the patient's in airway pressure, which is the difference between the peak pressure and the plateau pressure. The P reflects the pressure generated to overcome the resistance caused by the ventilator circuit, the endotracheal tube, and the patient's airways. PSV too is used for spontaneously breathing patients who have intact respiratory centers

In addition to the more commonly ordered modes of ventilation, several other modes have become available. Some of these modes are adaptations of CMV, continuous mandatory ventilation, i.e. a fixed pressure set by the operator. When the breath is triggered, the ventilator produces a rapid inspiratory flow to achieve the set pressure. When the desired pressure is reached and the lungs are filled, the flow decreases. The breath ends when the inspiratory time has passed. Volume delivery varies with the set inspiratory time and the patient's lung characteristics and whether the patient is actively inspiring.

High-Frequency Ventilation

Through high frequency oscillations, alveoli located closed to the airways are ventilated by convection, just as in conventional ventilation. [49, 30] The following additional mechanisms may be responsible. Gases move from one area of the lungs to another thanks to the difference in the compliance and resistance of lung regions; this is also called “out of phase ventilation”. This movement occurs through normal anatomic channels and it is enhanced when tissue is oscillating. [30, 46]

High-frequency ventilation (HFV) uses mandatory rates higher than normal and tidal volumes lower than normal; it is generally defined as any mode of ventilation that provides a mandatory rate of more than 100 breaths. Frequencies are usually given in Hertz (Hz).

There are lots types of HFV available, such as:

High-frequency jet ventilation (HFJV)

High-frequency oscillatory ventilation (HFOV)

High-frequency positive-pressure ventilation (HFPPV)

High-frequency flow interruption (HFFI)

High-frequency percussive ventilation (HFPV)

Probably the first two enumerated are more used.

High-Frequency Jet Ventilation

The high-frequency jet ventilation method (HFJV) usually uses a percutaneous transtracheal catheter. It uses rates between 1.7 and 10 Hz, with V moved smaller than anatomic dead space volume. In general, HFJV operates by passing gas from a high pressure source through a variable regulator that reduces the pressure to the desired working level. The gas then passes usually through a solenoid or a fluidic valve which governs the amount and duration of flow. The gas jet is then delivered through a specially made triple-lumen endotracheal tube. The jet stream does not completely reach the patient, because it exits the tube before to reach the tube's distal end. A second way to work with, is to use a small catheter inserted either through a conventional endotracheal or a tracheostomy tube. Early studies showed that the best position for the jet is dose to the proximal end of the trachea near the vocal cords. [49, 47] The operational principle of HFJV involves the delivery of short breaths or pulsations (20 to 34 msec) under pressure through a small lumen at high rates (5 to 11 Hz).

High-Frequency Oscillatory Ventilation

HFOV is the most widely used form of HFV in infants or pediatric patients. It is frequently used in neonatal and pediatric patients. HFOV has been shown to provide effective more lung recruitment than others to patients with ARDS. HFOV uses a reciprocating pump to generate an approximation of a sine wave. High-frequency flow can be produced even by ventilators to provide a similar effect. These ventilators are called "pseudo-oscillators”. With HFOV, pressure is positive in the airway during the inspiratory phase (forward stroke) and negative during the expiratory phase (return stroke). Therefore both inspiration and expiration are active, and bulk flow is produced. HFOV uses frequencies range of 1 to 50Hz.

High-Frequency Positive-Pressure Ventilation

HFPPV uses a conventional volume or pressure-limited ventilator with a low-compliance patient circuit. Mandatory rates are about 1 to 2Hz.

High-Frequency Flow Interruption

High-frequency flow interruption (HFFI) is similar to HFJV but differs in its technical design. In HFFI all mechanism interrupts a high-pressure gas source. A common mechanism is a rotating ball with a flow port in the center. The ball could be moved back and forced in the conduct by an electric motor at rates up to 200 cycles/min. As it moves, the ball interrupts the outflow of gas.

High-Frequency Percussive Ventilation

HFPV device presents the beneficial characteristics of both a conventional HFPP ventilator and a HFJ ventilator. It operates in such a way that high-frequency breaths can be provided at ambient pressures. High-frequency pulses (100 to 900 cycles/min) can also be superimposed on conventional positive-pressure breaths (5 to 30 cycles/min). In this case the pulses still occur, but the baseline rises as a positive-pressure breath is delivered. During inspiration, a diaphragm connected to the device fills with gas. This causes a block of the expiratory port, but at this time the jet manage the delivery of short pulses of gas. At the same time, a large amount of air is entrained through the now open inspiratory ports, so that flow to the patient is high. The large gas flow is due to the pressure gradient between the jet and the patient connector. As inspiration progresse, this gradient is reduced and so the flow is decreased too.

HFO valve can be set in order to implement a superimposed sinusoidal pressure wave as FOT requires. So whichever ventilation mode can be matched with forced oscillations. This is very useful both to have a unique device to measure and stimulate subjects, and to reduce set-up in NICUs and more generally ICUs, where there are already a lot of instruments and space needs to be saved. [18, 48]

2 – MATERIALS AND METHODS

In order to monitor lung mechanics during SLI, it is fundamental to have a set up easy to use in clinical practice.

FOT set up requires the generation of a sinusoidal pressure stimulus and flow and pressure measurements. All these task can be performed by a ventilator and therefore the ventilator used to apply SLI can be simultaneously measure the respiratory system impedance.

In particular, the generation of pressure stimulus can be achieved using HFOV with the desired frequency or controlling the expiratory valve, while flow and pressure are already measured by the all commercial ventilator thanks to a pressure meter and usually an anemometer.

Acutronic's FABIAN ventilator is an interesting device for FOT implementation. It is a neonatal ventilator with HFOV that measures pressure and flow at the mouth of the infant.

To permit to the clinicians to observe in real time R and X values with minimal ventilators' software changes, is necessary to connect it to an external device for impedance computation, data representation and storing.

This device has been thought to be a tablet which uses android operative system. The advantages of using a tablet are its dimension and its easiness of used. The main advantages of Android language are: its diffusion because of its complete portability, its low costs and its open source philosophy.

Therefore in this chapter the FABIAN HFOV Ventilators and hints of android programming are presented.

2.1 Hardware and Firmware

The aim of the following paragraph is to accurately describe all the parts used, both hardware and software.

2.1.1 Fabian's Acutronic jsc

Figure 2. 1 Acutronic's Fabian infant respiratory ventilator

The ventilator used in this Thesis has been produced by Acutronic Medical System AG (Aktiengesellschaft or "join stock company"). It is a prototype of the commercialized relative model "Fabian-HFO".

Technical Specifications

Before to explain in detail how the ventilator works and how it is composed, it has given a brief summary of its skills.

It is an electric and gas powered ventilator, with double tube of respiration. It is30 cm x 37 cm x 40 cm dimension and about 20kg of weight.

Manual breath can be extended 2-10s and a suited button disables the ventilation to let subject breathes spontaneously. HFO is a module which is supported by the device.

Accuracy of measurement in pressure, either Pmax, Pmean or PEEP is always ±4% with range 0 ÷ 100 cmH_2O or -10 ÷ 100 cmH_2O for PEEP.

Minute Neonatal Volume instead has a guaranteed accuracy of ±8% and a measurement range of 0 ÷ 999 ml.

Mechanical parameter of respiratory resistance has 0 ÷ 5000 cmH_2O of range, with an accuracy equal to ±8%.

To power the ventilator, it is necessary a compressor to refill it with gas.

Modes of Ventilation

Figure 2. 2 Modes of ventilation

The figure above shows several acronym which stands for the modes of ventilation affordable by Acutronic's Fabian. These acronyms have been described in the previous chapter (1.3.5). Sometimes, acronyms slightly change because of patents or preferences, anyway the differences are very easy to understand. A typical example is *BiPAP*, changed with *DUOPAP* in Figure 2. 2 or called Bi-Level by other brands. HFO mode is very useful and, as it will be further declared, it will be a fundamental mode used in the clinical setup.

Fabian components

The whole device can be observed through a modular sight. Different panels and several boards include orderly all the components which permit an excellent functioning of it.

Manifold Ports

This panel is collocated in the very lower front part of the device and as depicted in Figure 2. 3, it shows every airway connection. There are 4 ports. 3 of them (the ones positioned on the right) are always to be connected to the right hoses, while the very left one has to be connected to the set-up only if HFO ventilation is desired and turned on.

Figure 2. 3 Fabian's manifold port

The first manifold is the inspiratory tube port, as can be seen reading the picture. In case of CPAP only used, it is no necessary to match the second manifold (2), which is the one deputed to expiration. 3 is the port to sense pressure while the 4th is the port which permits to use High Frequency ventilation (High Frequency Oscillations).

Back Panel

Figure 2. 4 Back Panel

In Figure 2. 4 there are represented a lot of manifolds, to communicate data or simply to provide supply. Supply energy is provided by ports number 1 and number 10, which takes electricity from DC or AC external supply, respectively. Ports number 5, 6, 12 and 13 mainly concerns with ventilation because they respectively measure patient air flow, permit to communicate any alarm to central room and they take oxygen and medical and compressed air from hospital tubes.
In the upper-central part of the figure, 2, 3 and 4 manifold permit to obtain data or to update firmware. RS232 port is the one used to obtain data-packets of the values measured.

Back Panel - component side

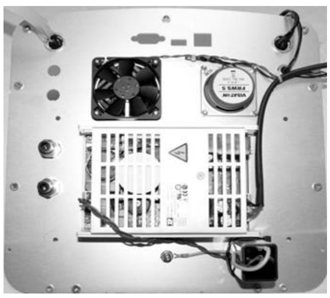


Figure 2. 5 Back panel – back side

As done before, it is shown the panel in both its sight; this is the internal view. In Figure 2. 5 it is easy to recognize the three part depicted: a board to control the loud speaker (which is positioned up-right) and to action the fan which disperse heat.

Fabian's Boards

In the internal part of the device there are different boards which executes different tasks

Mainboard – overview –

The Mainboard is the one where almost all tasks are related to. For instance the loudspeaker, all manifolds, the fan and the blender link to mainboard, but not all these part of functioning are directly controlled in this single module.

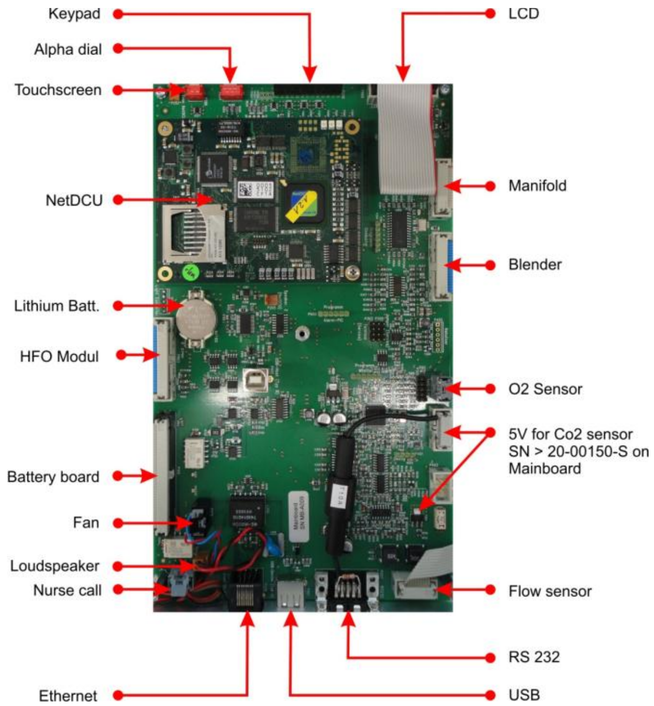


Figure 2. 6 Main Board

HFO Module: Board and Valve

HFO Module is very important to use a different way of ventilation. For the study in object, it has an additional importance as it can be used to generate forced oscillation to measure *Z*, as a superimposed waveform to calculate respiratory impedance. An easy schematic of HFO circuit is the one down depicted. It is easy to observe:

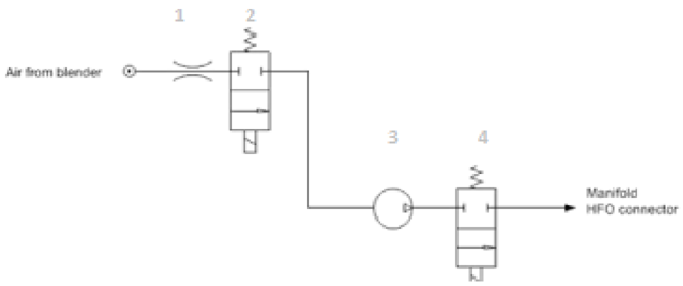


Figure 2. 7 Schematic of HFO functioning principle

As showed, air passes in through blender. Flow is restricted (#1) and firstly managed by #2 depicted valve; High frequencies are so created by the HFO generator (#3). Finally, the HFO valve (#4) manages definitely the turning on/off of the VM. The correspondent real part is represented in the Figure 2. 8 below

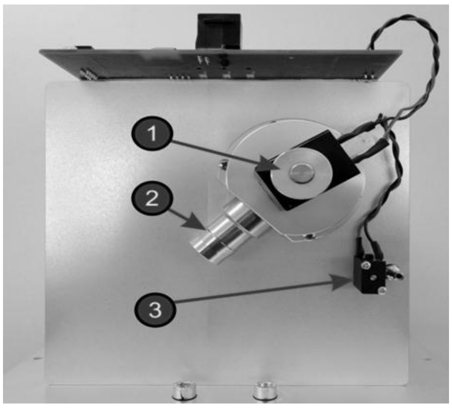


Figure 2. 8 HFO valve control

Specific parts are numbered and they are, in sequence, the HFP valve which controls open/close status, the outlet port and finally the connector to air bleed.

Drawing

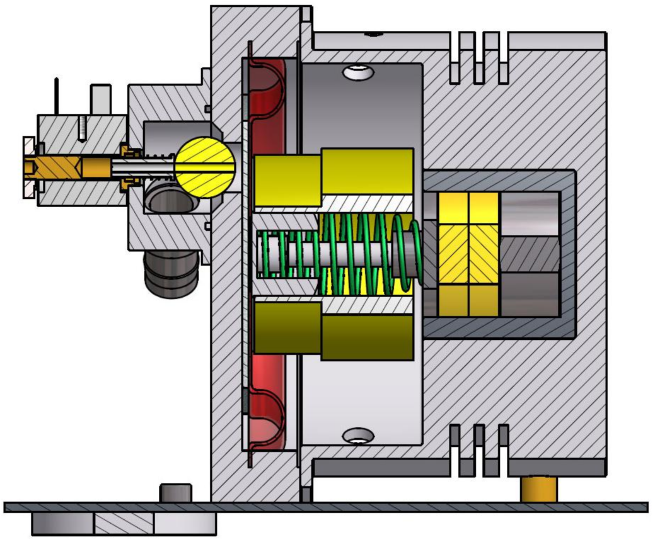


Figure 2. 9 Ventilator and HFO command draw

As explained in the previous chapter, the principle of functioning exploits the pneumatic power. Air enter the circuit thanks to a compressor or wall source (in hospitals) and is delivered to patient. Respiratory flow is produced by the piston (Figure 2. 9) in the centre of the draw and a valve in the upper left part handle the kind of flow and the expiration.

2.1.2 Anemometer

Figure 2. 10 Florian's anemometer

The hot-wire anemometer depicted in Figure 2. 10 above is comprehended in Fabian device. Its configuration is "constant temperature". CTA hot-wire anemometers (Constant Temperature Anemometers) works on the basis of convective heat transfer from a heated sensor to the surrounding fluid, the heat transfer being primarily related to the fluid velocity. The advantages of the CTA over other flow measuring principles is, above of all, the analogue voltage output, which means that no information is lost, and very high temporal resolution, which makes the CTA ideal for measuring spectra.

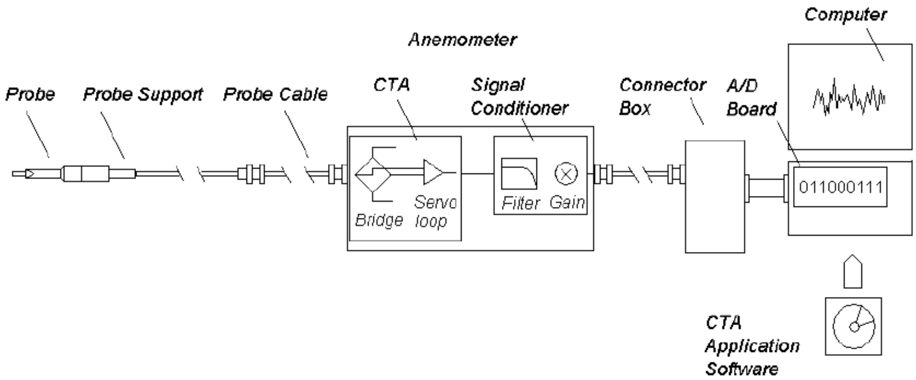


Figure 2. 11 Measurement schematic of anemometer

The measuring equipment constitutes a measuring chain, such as the one in Figure 2. 11. It consists typically of a Probe with Probe support and Cabling, a CTA anemometer, a Signal Conditioner, an A/D Converter, and a Computer. Very often a dedicated Application software for CTA set-up, data acquisition and data analysis is part of the CTA anemometer's chain. Actually, for the Master's Thesis in object, the measuring chain is not so explicit. This because signal conditioner to A/D board and comprehending CTA SW, is all included in Acutronic Fabian's boards (remember backpanel and the flowmeter connector -Figure 2. 4 -which comprehends the probe and its support and cable).

2.1.3 Samsung Galaxy Tab2 P5110

Figure 2. 12 Samsung's Galaxy Tab2 P5110

The matter a Tablet has been desired is that we wanted an independent tool, able to communicate with a ventilator and to start record of data and calculation of respiratory impedance. In fact, if we knew the data protocol of ay ventilator, we could access to data simply through an RS232-USB cable.

The tablet chosen for this Master's Thesis work has different particular aspects. First of all it manages host mode. The Host mode is very important because only *host* instruments can manage *devices* attached to them. Tablet usually are only "*device"*, i.e. when attached to PC they behave as classic USB pen-drive or whatever mass storage device; some tablet or mobile phone are instead capable to manage USB pen-drives themselves too (*host* mode). This skill has been carefully chosen because the need was to manage the respiratory ventilator as a device.

Samsung P5110 model was so decided to have all these peculiarities. It supports wifi connection very useful to connect to the net and download apps or send data via e-mail. It has neither 3G nor EDGE/UMTS connection, because it is not needed and moreover in ICU and NICU it is better to turn-off any radio-wave communication mode. Galaxy tab2 has 16Gb internal use memory and supports flash-memory. For this work, it has been equipped with a 4Gb micro SD-card. It is very useful to transfer data from tablet to PC in order to avoid even wifi connection for e-mail sending. The only one minus is that the tablet has not been rooted and thus it is impossible to create an Android application which automatically save data on ext-sd. Data will be only saved on "SD" partition, which however is like the "C:\" partition in windows OS.

System skills are very performing –considering that we are talking about a tablet- because of 1GHz Dual core processor and 1Gb RAM. 10.1 inches screen permit a good visualization of different graphs and. Operative system is the second to last newest, IceCream Sandwich4.0.3.

To run application, there are two main procedures: the first one it's the consumer way and it consists of registering an account and of downloading apk, i.e. Android applications only runnable whose code is not available. The second method is instead the developer one and it consists of linking the tablet to the PC and of forcing the run of an application created. The final result is however the same, thus tablet does not show any code but only the final result. To enable this useful way of installation, it is only necessary to select "USB debugging" option. It is the explicit request to run the developed application.

2.1.4 USB To Serial Converter

The Universal Serial Bus (USB in acronym) is an interface standard for communication between computers and peripheral devices. [42] Most devices connect to external peripherals through host ports provided by a USB processor chip. Anyway, it is also useful to convert data provided from different protocols such as the older ones; one of these is certainly the asynchronous Serial protocol RS232. Thus an USB to Serial adapter is a connector which permits to pass data from serial protocol, typically RS232, to Universal Serial Bus. The figure below represents different kinds of USB to Serial adapter. It comes in many models, sizes and shapes:

Figure 2. 13 Different kinds of USB to Serial adapter

The most obvious difference among them, is the physical size of the adapters; this is mostly due to the number of ports the particular adapter has. For example a one port adapter is much smaller than a four port adapter. As it is the major difference, mostly the physical size of the USB to serial adapter is actually the least important feature. Most USB to Serial adapters has a DB9 connector.

So, the real difference subsists in the circuitry inside. The main parts of the inside circuitry consists of two processor chips (chipsets): a USB to Serial processor chip which converts all the USB data to serial data and all the serial data to USB data, and a Serial driver chip, which on the contrary provides the power for the device connected to the serial port of the adapter. The chipset used in this Thesis' work, is FTDI.

The RS-232 protocol

RS-232 defines signals from two different perspectives: *data communications equipment* (DCE) and *data terminal equipment* (DTE). Actually DCE/DTE terminology evolved in the early days of computing when the common configuration was to have a terminal attached to a generic modem to enable communication with a mainframe computer in the next room or building. A person would sit at the DTE and communicate via the DCE. To explain it with different words, and to reduce confusion, the DTE was specified as a male DB-25 and the DCE as a female DB-25.

The principle behind RS-232 hardware can be summarized in 4 controls or requests; these signals are *request to send* (RTS), *clear to send* (CTS), *data terminal ready* (DTR), and *data set ready* (DSR). DTR/DSR enable the DTE and DCE to signal that they are both operational. The DTE asserts DTR, which is sensed by the DCE and vice versa with DSR. RTS/CTS enable actual data transfer. RTS is asserted by the DTE to signal that the DCE can send it data. CTS is asserted by the DCE to signal the DTE that it can send data. In the case of a modem, carrier detect is asserted to signal an active connection, and ring indicator is asserted when the telephone line rings, signaling that the DTE can instruct the modem to answer the phone. In RS-232, user data is sent as a time-series of bits. Both synchronous and asynchronous transmissions are supported by the standard. [42]

The Universal Serial Bus protocol

USB is limited by standard to no more than 5 meters of cable, thus favoring RS-232 when longer distances are needed. Both standards have software support in popular operating systems. USB is designed to make it easy for device drivers to communicate with hardware. However, there is no direct analog to the terminal programs used to let users communicate directly with serial ports. USB includes a protocol for transferring data to devices. This requires more software to support the protocol used. RS-232 only standardizes the voltage of signals and the functions of the physical interface pins. The design architecture of USB is asymmetrical in its topology. As shown in the figure below [Figure 2. 14], in fact it consists of a host, a multitude of downstream USB ports, and multiple peripheral devices connected in a star topology. A USB host may implement multiple host controllers and each host controller may provide one or more USB ports. USB devices are linked in series through hubs.

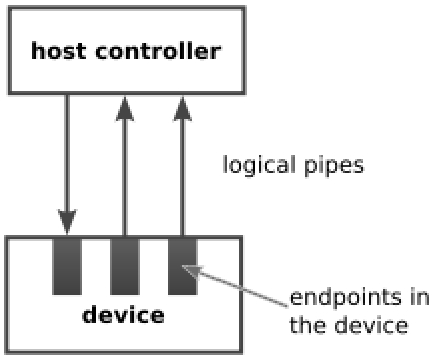


Figure 2. 14 USB transmission topology

USB device communication is based on pipes (logical channels). Each pipe is a connection from the host controller to a logical entity named as “endpoint”. Because pipes correspond 1-to-1 to endpoints, the terms are sometimes used interchangeably. Anyway, USB devices can have up to 32 endpoints, though they seldom have. Actually there is a slight but crucial difference between endpoints and pipes: the former is built into the USB device by the manufacturer and therefore exists permanently, the latter may programmatically be opened and closed. There are two types of pipes: stream and message pipes. A message pipe is bi-directional and is used to *control* transfers (in the next section, when Android’s code is presented, these concept obviously will be re-explained). A stream pipe is instead a uni-directional pipe connected to a uni-directional endpoint that transfers data using an *isochronous*, *interrupt*, or *bulk* transfer; For only bulk transfer will recur in the above-mentioned next section, it is the only one I am going to discuss about. Bulk transfer is large sporadic manner of communication which uses all remaining available bandwidth, but with no guarantees on bandwidth or latency (e.g., file transfers). If the direction of the data transfer is from the host to the endpoint, an OUT packet (a sort of a TOKEN packet) having the desired device address and endpoint number is sent by the host. If the direction of the data transfer is from the device to the host, the host sends an IN packet instead.

Endpoints are grouped into *interfaces* and each interface is associated with a single device function, which is enumerated whenever it is attached to an host.

Finally, the condition “to be an host” is fundamental because the host controller itself directs traffic flow to devices. No USB device can transfer any data on the bus without an explicit request from the host controller.

These last few lines are very important to introduce a new adapter, OTG USB adapter, which in fact can work only if the device attached supports the “host mode”.

2.1.5 OTG USB Host Converter

Figure 2. 15 OTG cable

The OTG converter is the one which permits the Galaxy tablet to control the communication with the ventilator. Thanks to this cable converter, the tablet gains the possibility to be the host (while usually it is the *device*, attached to the PC *host*) and the ventilator stand for the device. OTG converter has two ends: one male USB B to enter the female Samsung USB port, and one for USB A, thus a USB host receptacle. To repeat explain the concept, the Tablet has to have inside itself the host set.

2.2 Software

This paragraph *2.3* deals with hint of Object Oriented Programming and, subsequently, it describes Android programming Language. It has been chosen to quickly explain that, because object oriented programming is not so common. The record of respiratory data, and the calculation of respiratory impedance has been done thanks to an Android application. It is thus important to understand the rationale of the algorithm.

2.2.1 Introduction to Android Programming

2.2.2 Object Oriented Programming

Object oriented programming (OOP) is a programming paradigm using "objects", usually instances of a class - consisting of data fields and methods together with their interactions, to design applications and computer programs.

Programming features are, among all:

- data abstraction
- encapsulation
- messaging
- modularity
- polymorphism
- inheritance

An object-oriented program may be viewed as a collection of interacting objects, as opposed to the conventional model, in which a program is seen as a list of tasks (subroutines) to perform. In OOP, each object is capable of receiving messages, processing data, and sending messages to other objects. Each object can be viewed as an independent "machine" with a distinct role or responsibility. The actions (or *methods*) on these objects are closely associated with the object.

Programs are composed of self-sufficient modules (*classes*), each instance of which (*objects*) contains all the information needed to manipulate its own data structure (*members*). An object-oriented program will usually contain different types of objects, each type corresponding to a particular kind of complex data to be managed.

To simply understand the concept, an object can be thought as a real-world object or a real world concept such as a cat or a machine. A program might well contain multiple copies of each type of object, one for each of the real-world objects the program is dealing with. For instance, in a factory there are several machines, both equal and different, each one performing particular tasks. The same thing happens in programs.

Fundamental features and concepts

There are fundamental features that support the OOP programming style in most object-oriented languages:

- Dynamic dispatch – this feature distinguishes an object from an abstract data type (or module), which has a fixed (static) implementation of the operations for all instances
- Encapsulation (or multi-methods, in which case the state is kept separate)
- Subtype polymorphism
- Object inheritance (or delegation)

In a programming language, *encapsulation* is used to refer to one of two related but distinct notions, and sometimes to the combination of:

- a language mechanism for restricting access to some of the object's components.
- a language construct that facilitates the bundling of data with the methods (or other functions) operating on that data.

Some programming language researchers and academics use the first meaning alone or in combination with the second as a distinguishing feature of object oriented programming, while other programming languages which provide lexical closures, consider encapsulation as a feature of the language, in order to create inter-relations among objects.

In programming language theory, *subtyping* is a form of type polymorphism in which a subtype is a datatype that is related to another datatype (the supertype) by some notion of substitutability, meaning that program elements, typically subroutines or functions, written to operate on elements of the supertype can also operate on elements of the subtype. If S is a subtype of T, the subtyping relation is often written S <: T, to mean that any term of type S can be safely used in a context where a term of type T is expected. Because the subtyping relation allows a term to have (belong to) more than one type, subtyping is a form of type polymorphism, so it is properly called subtype polymorphism. Moreover, in object oriented programming subtyping is called polymorphism (see polymorphism in object-oriented programming).

OOP was developed to increase the reusability and maintainability of source code. Transparent representation of the control flow had no priority and was meant to be handled by a compiler. With the increasing relevance of parallel hardware and multithreaded coding, developer transparent control flow becomes more important, something hard to achieve with OOP.

Brief mention to Java programming language

Java is a programming language. It derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities than either C or C++. It is a typical language designed mainly for OOP, but with some procedural elements. Java applications in fact are compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, class-based and object-oriented language that is specifically designed to have as few implementation dependencies as possible. So Android takes after his program language base. They are thought to let application developers "write once, run anywhere" (WORA). The acronym means that code which runs on one platform does not need to be recompiled to run on another (portability).
Portability is achieved by compiling the code to an intermediate representation called Java bytecode, instead of directly to platform-specific machine code. Java bytecode instructions are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware. This is the only part which differs from a device to another.
There are anyway some contra. Programs written in Java have a reputation for being slower and requiring more memory than those written in C. Java does not support C/C++ style pointer arithmetic, where object addresses and unsigned integers (usually long integers) can be used interchangeably. The syntax of Java is largely derived from C++. Unlike C++, which combines the syntax for structured, generic, and object-oriented programming. All code is written inside a class. [50]

An example is the worldwide known “Hello world code” :

```
class HelloWorldApp {
   public static void main(String[] args) {
      System.out.println("Hello World!"); // Display the string.
   }
}
```

The keyword *public* denotes that a method can be called from code in other classes, or that a class may be used by classes outside the class hierarchy. The class hierarchy is related to the name of the directory in which the .java file is located. The second keyword you read is static which stands for a static method, associated only with the class and not with any specific instance of that class. Going on with the analysis, void has the usual and common meaning than in the others main program languages: the main method does not return any value to the invoker. The method name "main" is not a keyword in the Java language. It is simply the name of the method the Java launcher calls to pass control to the program. In fact the VM needs to be explicitly told which class to launch from. The main method must accept an array of String objects. Printing is part of a Java standard library: The system class defines a public static field called *out*. The out object is an instance of the PrintStream class and provides many methods for printing data to standard out, including printIn(String) which also appends a new line to the passed string.
The string "Hello, world!" is automatically converted to a String object by the compiler.

2.2.3 Eclipse

Figure 2. 16 Eclipse logo

Eclipse is a software development environment usable for different languages, comprising an integrated development environment (IDE) and an extensible plug-in system. It can be used also to develop applications in Java. Development environments include the Eclipse Java development tools (JDT) for Java.
The Eclipse Platform uses plug-ins to provide all the functionality system. This is a big difference among other applications, in which functionality is hard coded. The plug-in architecture supports writing any desired extension to the environment, such as for configuration management. In this respect, all features are "created equal". The Eclipse Software Development Kit (SDK) includes the Eclipse JDT, offering an IDE with a built-in incremental Java compiler and a full model of the Java source files.

2.2.4 Android

Figure 2. 17 Android logo

The main software used in this Master's thesis work is Android. To develop Android application, is necessary to install in Eclipse plug-in IDE the Java Development Kit (JDK) - to be noticed: a synonym of JDK is SDK which is the acronym of Software Development Kit. Eclipse helps the programmer auto-compiling part of code, creating from its own formatters and giving the possibility to program graphically. Android is based on Java program language and exploits Eclipse IDE. "Google's Android platform has taken off. Since the first *Android Application Development For Dummies* book was released, Android has gained a tremendous amount of traction in regard to market share in the smartphones space" [51] This citation testifies how popular began Android language. Android SDK permits to develop a software exploitable by whichever hardware which support Android language. Android is a Linux-based operating system designed for touch screens mobile devices such as smart phones and tablet computers. From the development point of view, the core of the writing is a customized version of Java. Android consists of a kernel based on the Linux kernel, with libraries and APIs running on an application framework which includes Java-compatible libraries. Android uses the Dalvik virtual machine with just-in-time compilation to run Dalvik dex-code (Dalvik Executable), which is usually translated from Java bytecode. The main hardware platform for Android is the ARM architecture (a family of computer processors). Applications are usually created using the Android Software Development Kit. The run of applications is in a sandbox, i.e. an isolated area of the OS that does not have access to the rest of the system's resources. The user has to explicitly permits the access to the system when the app is installed.

Brief mention to Dalvik Virtual Machine

Dalvik Virtual Machine, as mentioned before, is the VM which supports Android. Unlike Java VMs, which are stack machines, the Dalvik VM uses a register-based architecture. A suited tool called dx is used to convert some Java *.class* files into the *.dex* format. A singe dex file contains several classes and constants, mentioned only once to save space memory. Moreover, Java bytecode is converted too into an alternative instruction set. For instance, an uncompressed *.dex* file is typically a few percentage points smaller in size than a compressed *.jar* (Java Archive) derived from the same *.class* files. The Dalvik executables may be modified again when installed onto a mobile device. In order to gain further optimizations, byte order may be swapped in certain data, simple data structures and function libraries may be linked inline.

Android Code Architecture and User Interface

Application Components

Application components are the essential building blocks of an Android application. Each component is a different point through which the system can enter your application. Not all components are actual entry points for the user and some depend on each other, but each one exists as its own entity and plays a specific role-each one is a unique building block that helps define your application's overall behavior.

There are four different types of application components. Each type serves a distinct purpose and has a distinct lifecycle that defines how the component is created and destroyed. When the system starts a component, it starts the process for that application (if it's not already running) and instantiates the classes needed for the component.

Here are the four types of application components:

Services

A service is a component that runs in the background to perform long-running operations or to perform work for remote processes. A service does not provide a user interface.

Content providers

A content provider manages a shared set of application data.

Broadcast receivers

A broadcast receiver is a component that responds to system-wide broadcast announcements. Many broadcasts originate from the system—for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured.

Activities

An activity represents a single screen with a user interface. For example, an email application might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. Although the activities can work together to form a cohesive user experience, each one is independent of the others.

Typically, one activity in an application is specified as the "main" activity, which is presented to the user when launching the application for the first time. Each activity can then start another activity in order to perform different actions. Each time a new activity starts, the previous activity is stopped, but the system preserves the activity in a stack (the "back stack"). When a new activity starts, it is pushed onto the back stack and takes user focus. The back stack abides to the basic "last in, first out" stack mechanism.

When an activity is stopped because a new activity starts, it is notified of this change in state through the activity's lifecycle callback methods. When the activity resumes, even pre interrupted actions will resume. These state transitions are all part of the activity lifecycle.

Creating an Activity

To create an activity, it is necessary to create a subclass of Activity (or an existing subclass of it). The subclass, permits to implement callback methods that system calls when the activity transitions between various states of its lifecycle, such as when the activity is being created, stopped, resumed, or destroyed. The two most important callback methods are:

-onCreate()

Necessary to implement. It is called once, when the activity is created. The graphical elements are set thanks to *setContentView*().

-onPause()

The system calls this method as the first indication that the user is leaving your activity (though it does not always mean the activity is being destroyed). This is usually where you should commit any changes that should be persisted beyond the current user session (because the user might not come back).

An activity can exist in essentially three states:

Resumed or Running

The activity is in the foreground of the screen and has user focus.

Paused

Another activity is in the foreground and has focus, but the activity in object is still visible. That is, another activity is visible on top of this one and that activity is partially transparent or doesn't cover the entire screen. A paused activity is completely alive (the Activity object is retained in memory, it maintains all state and member information, and remains attached to the window manager), but can be killed by the system in extremely low memory situations.

Stopped

The activity is completely obscured by another activity (the activity is now in the "background"). A stopped activity is also still alive but it is not attached to the window manager. However, it is no longer visible to the user and it can be killed by the system when memory is needed elsewhere.

If an activity is paused or stopped, the system can drop it from memory either by asking it to finish (calling its finish() method), or simply killing its process.

The down depicted figure represents a general schematic of an activity lifecycle.

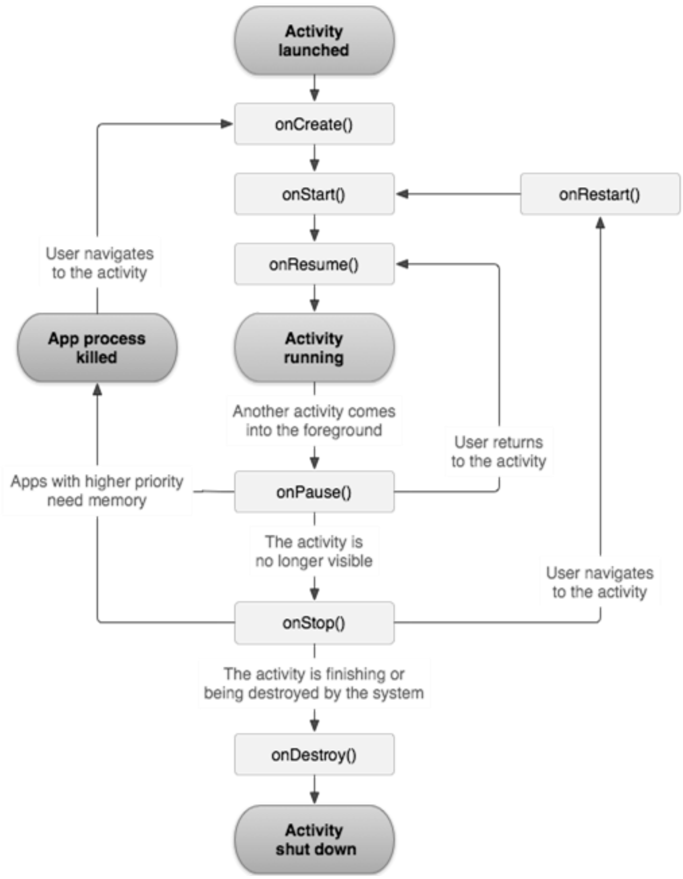


Figure 2. 18 Activity lifecycle

The starting point is the central upper one: "Activity launched". The activity starts for the first time – it is launched. Layout elements are created and started. The first "onResume()" is actually un-useful but it serves to give cyclic properties to the scheme. As easy to read, application can be stopped, paused or killed. All these actions are done directly by the user, or after a while by the OS, especially when another activity would be in conflict and this one in object is no longer in the foreground.

The user interface for an activity is provided by a hierarchy of views—objects derived from the View class. Each view controls a particular rectangular space within the activity's window and can respond to user interaction. For example, a view might be a button that initiates an action when the user touches it.
In Eclipse plug-in, the Package Explorer gives a global view of Projects to be developed.

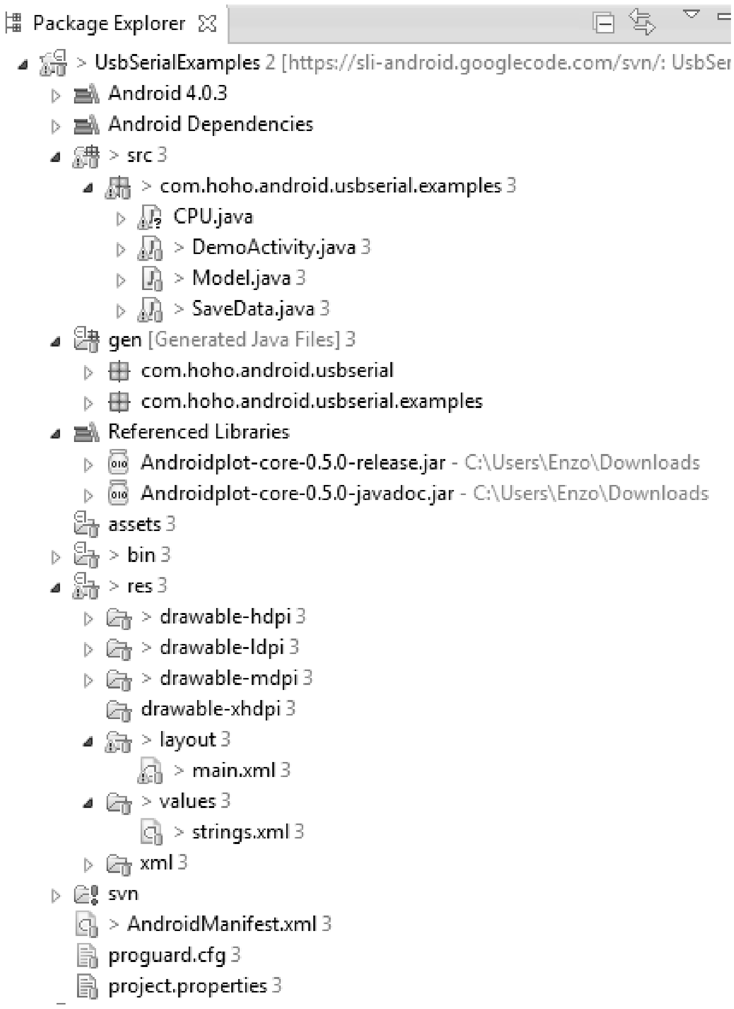


Figure 2. 19 Eclipse explorer view

Make reference to some folders - Figure 2. 19- which require more attention than other:

- svn\
- Referenced Libraries\
- gen\
- res\
- layout\
- src\

Svn\ is a backup folder, actually very important in case the app is being developed in team. It means, every registered user can modify the code, and it is possible both to merge the different parts or to restore old lines.

Referenced Libraries\ is a sort of archive, or better a “Java archive”, or JAR, which attaches other source codes necessary to perform some actions.

Gen\ folder contain auto-generated files and usually it is not necessary to open them.

Some kind of files deserve particular attention because of their import ace and their peculiarities.

The Layout File

Res\ folder contains different other folders such as the application icon, the layout and some strings.

Layouts are views derived from ViewGroup that provide a unique layout model for its child views, such as a linear layout, a grid layout, or relative layout. The most common way to define a layout using views is with an *XML layout* file saved in application resources. This way, you can maintain the design of your user interface separately from the source code that defines the activity's behavior. You can set the layout as the UI for your activity with setContentView(), passing the resource ID for the layout. However, you can also create new Views in your activity code and build a view hierarchy by inserting new Views into a ViewGroup, then use that layout by passing the root ViewGroup to setContentView().

Src\ is the folder which contains the activities. The main activity needs to be declared in the **manifest**, in order to be accessible by the system. To declare the activity, the manifest file needs to be opened and the activity in object has to be added as a child of the <application> element.

The Manifest File

Before the Android system can start an application component, the system must know that the component exists by reading the application's *AndroidManifest.xml* file (the "manifest" file). Every application must declare all its components in this file, which must be at the root of the application project directory. The manifest does a number of things in addition to declaring the application's components, such as:

- Identify any user permissions the application requires, such as Internet access or read-access to the user's contacts.
- Declare the minimum API Level required by the application, based on which APIs the application uses (API is the acronym for Application Programming Interface).
- Declare hardware and software features used or required by the application, such as a camera, blue-tooth services, or a mult-itouch screen.
- API libraries the application needs to be linked against (other than the Android framework APIs), such as the Google Maps library.

The very first lines of Manifest file are the following – it is quite completely automatic in its compilation:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.hoho.android.usbserial.examples"
    android:versionCode="1"
    android:versionName="1.0" >
```

Intents and Intent Filters

An Intent object is a bundle of information. It contains information of interest to the component that receives the intent - as the action to be taken and the data to act on – and even information of interest to the Android system (such as the category of component that should handle the intent and instructions on how to launch a target activity). Three of the core components of an application (activities, services, and broadcast receivers) are activated through messages, called *intents*. Intent messaging is a facility for late run-time binding between components in the same or different applications. The intent itself, an Intent object, is a passive data structure holding an abstract description of an operation to be performed - or, often in the case of broadcasts, a description of something that has happened and is being announced.

The Android system finds the appropriate activity, service, or set of broadcast receivers to respond to the intent, instantiating them if necessary. There is no overlap within these messaging systems: Broadcast intents are delivered only to broadcast receivers, never to activities or services. An intent passed to *startActivity()* is delivered only to an activity, never to a service or broadcast receiver, and so on.

This document begins with a description of Intent objects. It then describes the rules Android uses to map intents to components — how it resolves which component should receive an intent message. For intents that don't explicitly name a target component, this process involves testing the Intent object against *intent filters* associated with potential targets.

3 - LUNG MECHANICS MEASUREMENT SYSTEM

In this Chapter the set up developed in this thesis will be presented.

The setup comprises mainly the Fabian HFOV ventilator and Tablet with an android platform ,which have been described in the previous chapter,

The ventilator provide patients with pressure ventilation and, at the same time, generates the FOT pressure stimulus thanks to HFOV. Measurements of flow and pressure are performed by the ventilator too.

In order to use this ventilator for FOT measurements pressure and flow data provided from the ventilator should be accurate enough and perfectly synchronized. Therefore it was necessary to downgraded the threshold usually applied by the ventilator to avoid noise around zero flow as all the flow values under this threshold were forced to zero. Moreover pressure and flow data, delivered from ventilator to tablet, are not perfectly synchronized in time, because of the two different principles of sampling of flow and pressure. It is thus required a correction. Finally a communication protocol has been defined to send data from the ventilator to the Android application.

The Android application created and run on the tablet, acquires data packets from the ventilator. Packets are unbounded in flow data and pressure ones. They are used to calculate respiratory resistance and reactance, using the least mean squared algorithm. Every track is plotted run time and saved in a file which is automatically created at each record.

3.1 Setup overview

The setup crated is depicted in the figure below -Figure 3. 1:

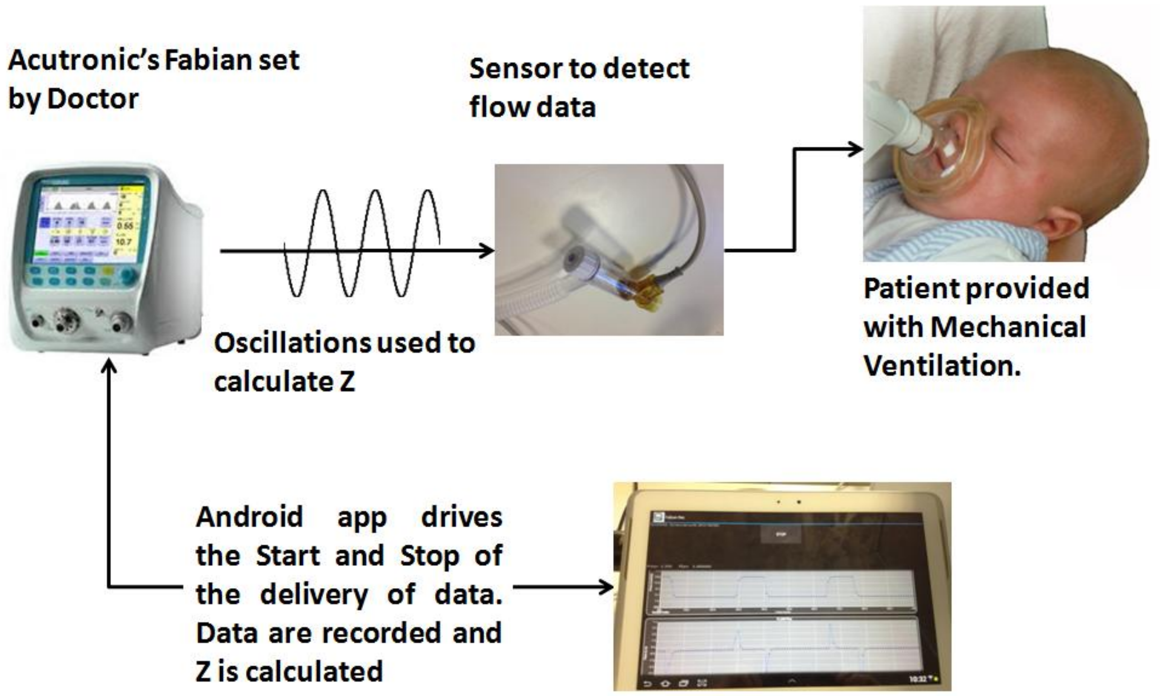


Figure 3. 1 Developed Setup. It is the Setup used in NICU to investigate the recruiting manoeuvre. The Ventilator exploits HFO mode of ventilation and applies oscillations to the patient. data are recorded by the Tablet. The application calculates respiratory impedance, thanks to the oscillations generated.

As easy to understand, the infant ventilator provides patient with mechanical ventilation. The amount of pressure delivered, is combined with oscillations generated internally by the ventilator. These oscillations permit to calculate the respiratory impedance Z_{RS}. Data are sensed by a pressure proximal sensor – an hose not depicted – and by the yellow anemometer showed. The setup includes a Samsung's Tablet whose specific application, created for this Thesis, communicates with the ventilator and decides the start and the stop of delivery of data. Data of pressure and flow are recorded in a specific file. They are eventually corrected - as it will be explained in the next Chapter - and thanks to an algorithm, the Real and Imaginary values of Z are calculated and saved. To be noticed the *inflation valve,* linked to the yellow anemometer. It is fundamental to execute the SLI technique. The Figure 3. 2I s representative of the manoeuvre.

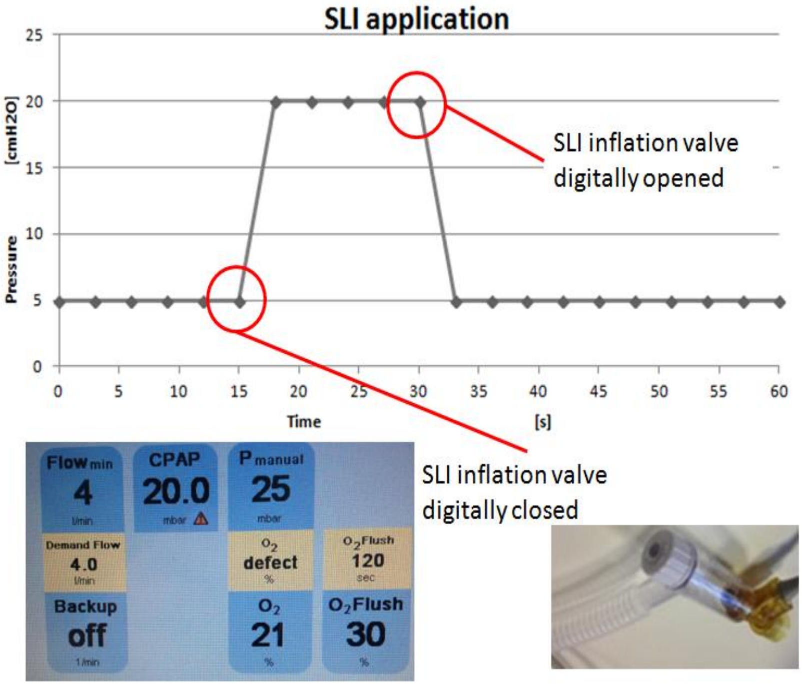


Figure 3. 2 SLI manoeuvre. It represents the changes in pressure at mouth. The inflation valve ensures a fixed PEEP when let opened, and the fixed value of CPAP, as PIP, when digitally closed.

The fast change of pressure, is allowed by a specific SLI respiratory circuit. It has an had-adjustable valve whose rotation creates a leakage in the circuit. The leakage is proportional to the rotation. In this way, the ventilator parameters are set in order to reach the maximum value – 20 cmH_2O. The valve is regulated in order to have a loss of 15 cm H_2O. when the user decides to apply SLI, his finger closes the valve. No loss is left and the pressure rises to the peak value. The SLI valve appears in the right down part of the image: the little hole in the middle of the white cap, it the one which let airflow exit the circuit. Rotating the white cap, the user is able to control the desired leakage.

In the next paragraphs, there are described the changes applied to the ventilator and the development of the application.

3.2 Fabian HFOV Firmware Changes

In order to permit the FOT measurements, something has been changed in agreement with Acutronic AG. The changes concerns mainly with data elaboration and filtering.

3.2.1 Threshold Limitation and Filtrage

Original firmware version did not care about small delays of pressure and flow, because ventilator only need to monitor the parameters delivered. Moreover, original algorithm had a threshold which did not permit to observe small changes, neither in pressure nor in flow. In the next figure, Figure 3. 3 it is in fact represented the comparison between the pressure signal sensed by Fabian – in blue - and by another pneumotacograph, reliable and chosen as gold-standard – colored in red. Because of FOT, a smaller waveform is superimposed and it is necessary the ventilator to be sensitive as much as possible.

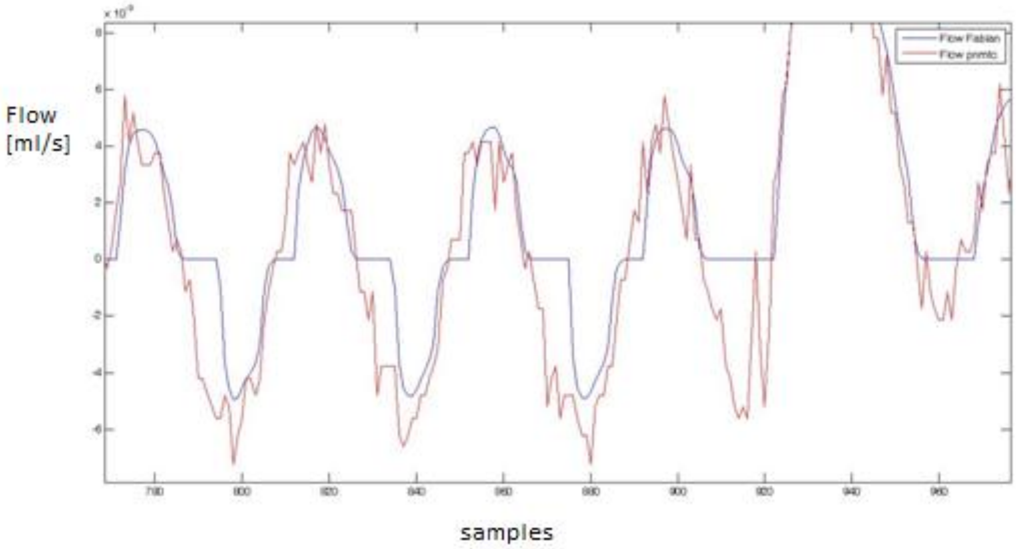


Figure 3. 3 Comparison in flow sensed by Fabian's sensors – in blue -and by gold standard's ones – in red. The threshold do not permit the ventilator to be as sensitive as possible to use FOT.

According to Acutronic AG, the threshold has been changed from 120ml/min (the costumed one), to **20ml/min**. The constructor referred that a minimum threshold has to be maintained because the need to avoid noise.

Proximal pressure is both hardware and software filtered. Better, in HFO VM only hardware filter is applied. It is a low-pass filter with time constant equal to 0,33 ms.

Software-filtering is instead applied to all modes except HFO. The output averages new and old sample.

$$Out = {}^{1}/_{8}\,Val_n + {}^{7}/_{8}\,Val_{n-1}$$

[Eq 3.1]

Sample rate is 1KHz and time constant 8 ms.

Patient Flow at Y adapter is one more time both hardware and software filtered. Hardware-filtering concerns with all the modes of ventilation and its time constant is 0.104 ms.

Software-filtering is used in all modes except HFO.

$$Out = {}^{1}/_{4}\,Val_n + {}^{3}/_{4}\,Val_{n-1} \qquad \text{[Eq. 3.2]}$$

Sample rate is 1 ms and the resulting time constant is halved: 4 ms.

3.2.2 Firmware Data Communication

Acutronic's Fabian stores data 200 times per second (200Hz); data are a couple of values representing flow and pressure. Flow data are measured by the hot-wire anemometer while pressure data are sampled as close as possible to subject's mouth and are passed through a hose in Manifold panel as described in paraghaph 2.1.1. Data are bounded in packets; each packet contain ten couple of values and it is so composed:

```
#press1|flow1|press2|flow2|press3|flow3|press4|flow4|.....press9|flow9|press10|flow10
```

Figure 2. 20 Example of packet data communication binding

"#" symbol is a sort of header file and precedes twenty values divided by a bar, "|". Each packet occurs at 20Hz. Because of developer's choice, data are sent to Serial output port as *char*. This implies that high numbers are sent as several chars, on the contrary they are sent as few chars. The result is that the number of byte changes each packets and the user needs to have a non fixed buffer.

Moreover, Serial communication does not happen always but it is necessary to define a *start* or a *stop*. In this perspective, it is only necessary to write a 'M' char or a 'Q'. This letters stand for "get continuous Monitor data" and "Quit continuous data sending", respectively. COM-Port skills are:

COM baudrate	115200
Data bits	8
Stop bits	1
Parity	None
Control Flow	XON/XOFF

3.3 Android Application

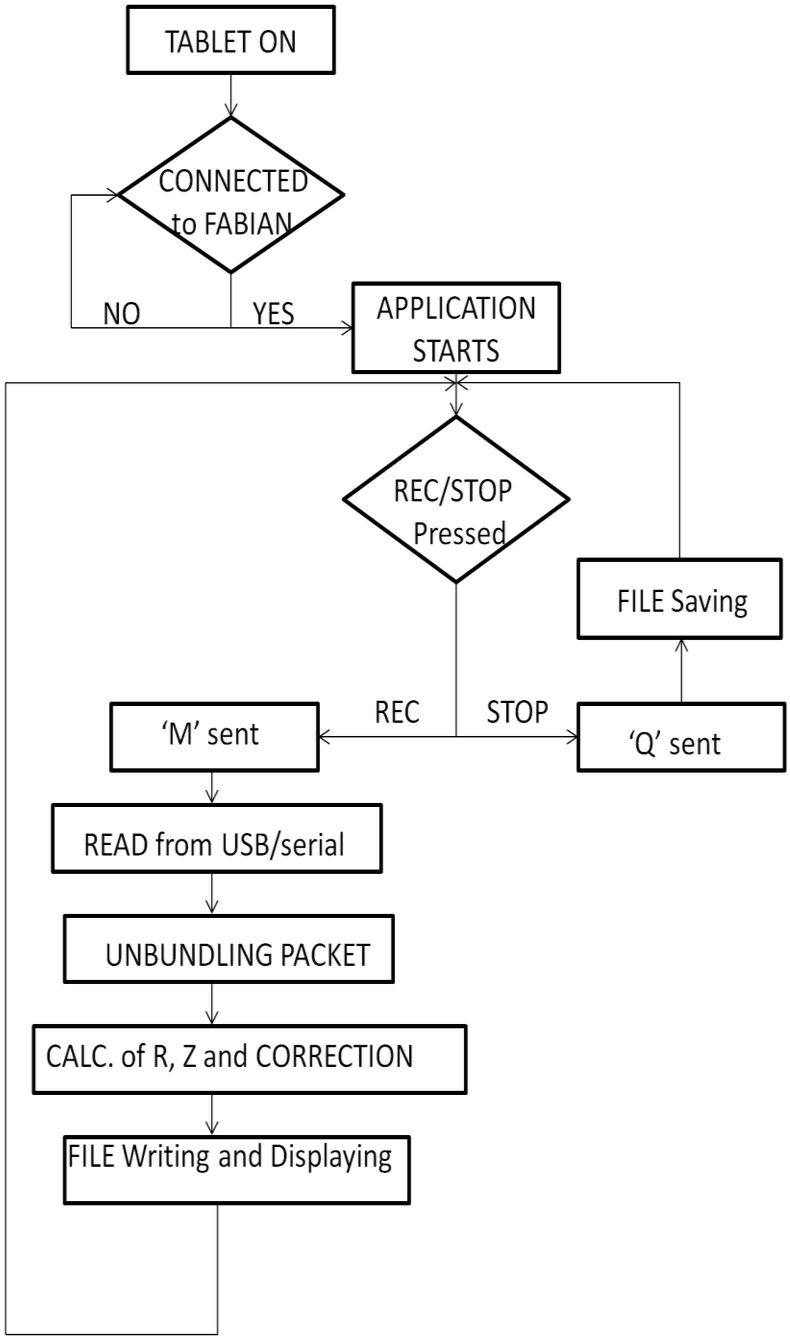


Figure 3. 4 Finite states schematics

Figure 3. 4 represents the global functioning of the application. In these first lines it is explained what the program does; after that it will be explained how it works. When the Tablet is turned on, to run the application is not necessary to click the application's icon. In fact whether the ventilator is connected to the tablet, a pop-up directly appears asking to let the application start. When the program is active, the user needs only to press the button

"REC" in order to start recording. Thus the communication between the infant ventilator and the tablet starts: an 'M' character is sent to the ventilator and after that data are collected. Pressure and flow data are displayed and used to calculate both the respiratory resistance and reactance. All data are saved in a file named with data and time of the start. When the user wants to stop the record, he presses one more time the button, in this case, the consequence is a stop: a 'Q' is sent and every acquisition is stopped.

Figure 3. 5 Record of data

Going on with a more detailed explanation, not only the principle of functioning is explained, but even how the steps are performed. The application communicates with Fabian's ventilator through serial communication. Serial port waits a start signal, 'M', to start sending pressure and flow data, bounded as explained before; at the same time, with the hearing of the 'Q' character, serial communication stops.

As the lecturer can see from **Errore. L'origine riferimento non è stata trovata.**, there are several graphical icon to explain:

Figure 3. 6 Application layout

If the script “Device connesso” – make reference to Figure 3. 5 - is present in Status Text view, there is not any problem in the communication and it is possible to start recording data. To do this, the operator need simply to press the touch screen on the button “Rec”. The click of the button implies, as said, the sending of ‘M’ char in output - with respect to Tablet - and to the consequent listening at input data from the ventilator. When communication starts, some icon changes because of data incoming from the respiratory ventilators.

First of all it is shown the name of the file where the Tablet’s *file pointer* is writing (as depicted in left-top). The button’s text icon changes suddenly in “Stop” because now any other press will lead to the stopping of communication and thus to stop the recording.

When packets are obtained, they are unbounded and pressure and data flow are written in the *.txt* file above mentioned, in a single line as in a table. Thanks to the Least Mean Squared algorithm, pressure and flow are used to obtain in real time the respiratory resistance and reactance, which are written in two other column.

To be noticed that LMSQ method will be explained in detail in the next chapters lines.

Whenever a dataset of these 4 kind of values is created, four chart are updated. The chart can be seen only in couple in a single sight, but thanks to a scroll-touch the operator can observe whichever he prefers. In the end, pressing the “Stop” button, ‘Q’ is sent, the file is closed, the communication stopped and the two text icons described initially, change one more time to the initial state.

3.3.1 Code Architecture

Fabian Rec app is stored in a single Android Project, which comprises the different files already explained, and other ones. For instance, to perform graph plotting, two *.jar* files have been attached. These codes are not usually stored in Android SDK, that's why it is necessary to declare it in the Referenced Libraries folder. "Androidplot" is so an open source code downloaded from www.androidplot.com and used in this Master's Thesis work. Another very useful code has been downloaded from googlecode.com: UsbSerialLibraries, in order to easily manage the USB to RS232 protocol communication.

Fabian Rec has been designed in order to perfectly fit in *MVC style*. MVC is the acronym for Model, View, Controller, very used by developers, to maintain the order in programs - Figure 3. 7

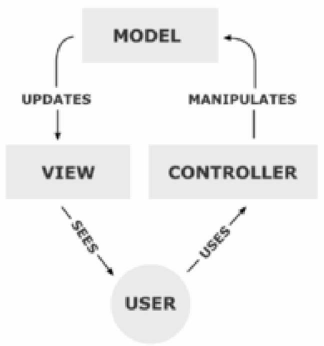


Figure 3. 7 Model-View-Controller rational

The *model* consists of application data and business rules; the *controller* mediates inputs, managing commands for the model or view. The *view* is finally any output representation of data, such as a chart or a diagram. The central idea behind MVC is code reusability and separation of concerns. Unfortunately there was a problem in invoking some arguments and libraries methods and moreover it was dangerous to let manage graph drawing not to Main file because the operations to perform was a lot and the risk was to stuck the app. Thus the MVC pattern was abandoned at least in part. These two .java file was put in DemoActivity file.

There are different files which construct all together the application, from both the graphical and the mathematical point of view.

Excluding the Manifest file, accurately described previously, 5 other files – a layout and the 4 mentioned files- deserve to be explained:

- Main.xml
- DemoActivity.java
- Model.java
- CPU.java
- SaveData.java

Main.xml

This *.xml* file is the one already described which guarantees graphical layout skills.

The layout chosen is linear in the first upper part, with a text declared to communicate status and filename to user. Next to the text view, there is the button here declared:

```
<Button
1       android:id="@+id/button"
2       android:layout_width="wrap_content"
3       android:layout_height="wrap_content"
4       android:layout_gravity="right"
5       android:height="80dp"
6       android:onClick="recordData"
7       android:text="@string/button_go"
8       android:visibility="visible"
9       android:width="200dp" />
```

The object *button* is declared in the first line (#1), while the line 6 explicits the method to call whenever the object is clicked (or touched). The other lines express dimension and physical characteristics, except the line number 7 which assigns a *string value*, or better a name, to the object.

The rest of the space is divided into a sort of table: 4 graphs are showed in a vertical view; thanks to a scroll, only two of them are observable in a single sight.

```
<com.androidplot.xy.XYPlot
        android:id="@+id/plot1"
        android:layout_width="1200dp"
        android:layout_height="290dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="10dp"
        title="PRESS Plot" />
```

The same code is repeated for the other three graphs.

The four other *.java* codes are the attempt to order in different files, the single program. The following four *.java* files are linked one to another, as explained before in the MVC philosophy. The following graph - Figure 3. 8 - explains how the file are nested.

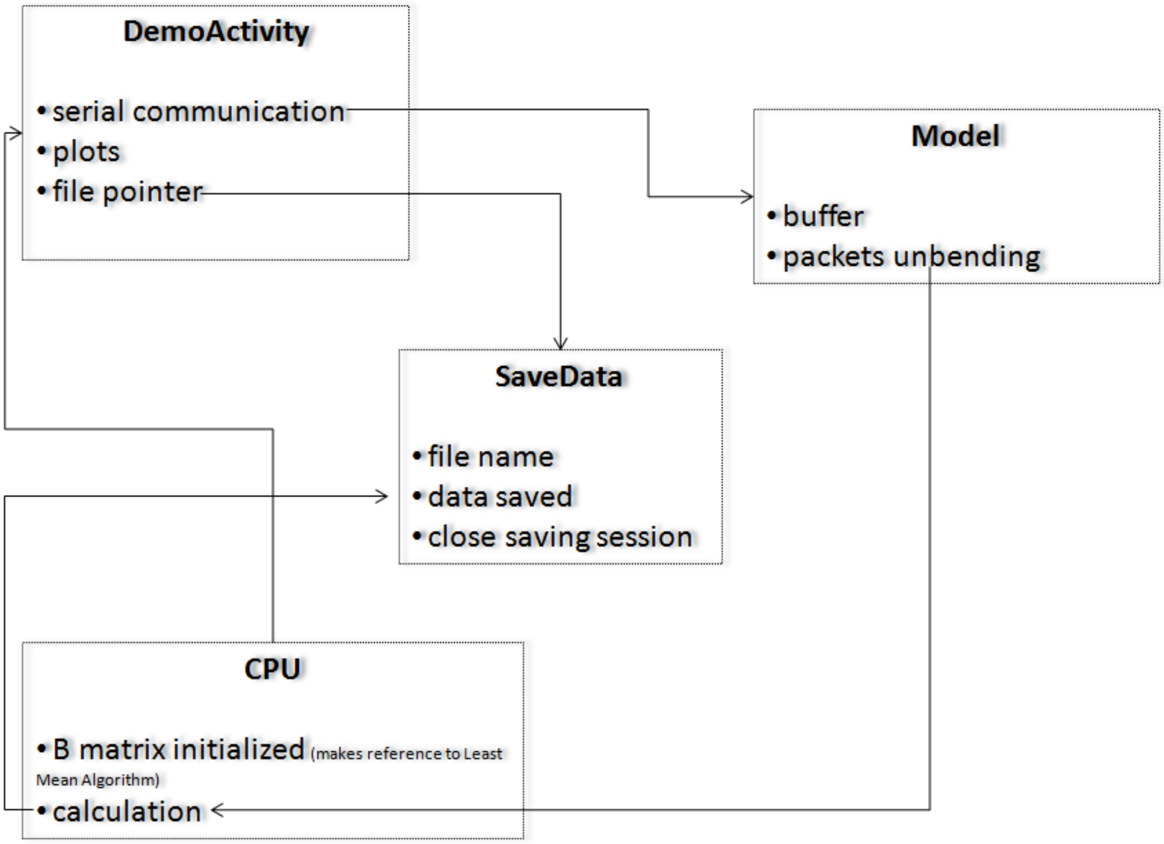


Figure 3. 8 Code Architecture

DemoActivity.java

If the application was a C program, this file would be its main. When application starts, a text tells the operator whether the device (the respiratory ventilator) is connected. If yes, the operator can press the button "rec". when he does this, something happens:

- communication starts because of output stream of 'M' starting character and reading (listening) of incoming bytes
- the button changes its script in "Stop", because now the communication and the consequent recording are running and it is possible to stop
- the script which told the status of the connection disappears and appears on the contrary, the root name of the file where data are written

The file root express in a numeric format the year, the month, the day, the minute and the seconds where the record has started: for instance if the starting date and time are 11th January, 2013 at 21 and 15, the format will be “201301112115-part…” so attached to the numeric format there is a string which contain a numeric incremental number: part0, part1,…part3. This, because the file will be broken into several file every 7 minutes, in order not to have huge data to observe in a single sight.
Going into the written code, initially there are declarations and initializations. After, it is decided what fragments to create at the very first run of the application, and what to do when it is on pause and when it is resumed.
DemoActivity creates a *SerialIoManager*, an entity which manages serial communication.
Data arrive already unbounded thanks to *Model* and *CPU* files. *CPU* offers Resistance and Inertance values too. *DataActivity* displays these data even on graphs (one for each values: Pressure, Flow, Resistance and Inertance). Because of difficult in updating data, it has been chosen to display values with a demultiplexing value: 5. Thus, instead of data @200Hz, the sampling frequency is 40Hz. Make reference that according to Shannon Sampling Theorem, 5Hz sinus waves used in FOT measurement is not equivocated. User is able to observe in a single sight, 2.5 seconds of record for each one of the four tracks.

Model.java
According to *MVC* philosophy, while the communication is handled by *DataActivity, Model* manages application data. A buffer is created. Because data are sent in packed headed with the symbol #, the algorithm tries to find this char symbol. When it finds it, and whether it finds another one, it starts to unbound the values. It is used the method *.split* to divide values before and after every | found. It is thus created a suited type called *dataPacket,* which have a couple of values: pressure and flow. The method described is very useful in order to avoid data missing: does the lecturer think about the automatic break of file and start of the second part (“-part1”); it could happen from whatever values, being pressure or flow. In this way, only at the starting of communication, and stopping it, maximum 20 data can be lost, but the probability of mistakes is then nullified.
Data are opportunely converted as desired in order to present L/min flow ventilation and 0.1 cm H_2O pressure. After, they are passed to *Save* and *CPU.java*.

CPU.java

This *.java* file receives data of pressure and flow and calculates impedance values according to LMSQ method. LMSQ method will be accurately explained in the next chapter.

To calculate this, it has been created a list of a fixed number of data (40) and every time a data is sampled, the related R and X values are calculated and transmitted to *SaveData* file. The number of data derives from the quotient between the sampling frequency (200Hz) and the frequency of the stimulus (5Hz):

$$window = \frac{f\ sample}{f\ signal} = \frac{200\ Hz}{5\ Hz} = 40$$

R and X values are also corrected according to the correction factor calculated during in vitro test - make reference to Chapter 3.

SaveData.java

The last *.java* file to describe is the one which handle the saving of data and, previously, the creation of a folder and of a path where to save them. The folder is changed every day because it is created as a date format – similar to file name, but only with year, month and day info. In this java file it is also implemented a timer event, a sort of interrupt, which changes the path where to save incoming data, maintaining the root file (refer to *DemoActivity*) but changing the suffix with *part1, part2…* the class *write*, implemented in *SaveData* permits the effective saving of data; it is however called by *DataActivity.java* whenever new values are read from serial bus.

3.3.2 The Least Mean Squares Method and the Calculation of Respiratory Impedance

The estimation of respiratory impedance is performed using the Least Mean Square Method (LMSM). Sinusoidal coefficients are thus calculated. The algorithm is implemented on Android application. the same algorithm has been used in offline analysis using a Matlab suited function, to validate the application.

The advantages of computing impedance in this way are various, such as:

- possibility of intra-breath analysis of impedance's changes.

- it is not necessary the assumption that the system under analysis has achieved a dynamic oscillatory steady state.
- high temporal resolution.

Ideal sine waves are fitted on the observed flow and pressure oscillation to find the Fourier coefficients (or equivalently the phase and amplitude of the sine) that minimize the least squares difference between these two signals. From these coefficients is possible to obtain the impedance.

The ideal sinus can be expressed as down showed:

$$S(t) = n(t) + a_0 + a_1 \cos 2\pi f t - b_1 \sin 2\pi f t$$

[Eq. 2. 1]

This equation considers a general sinusoidal function at a known frequency f as the result of a weighted sum of the orthogonal bases sine and cosine, at the already mentioned frequency. a_1 and b_1 represent the weights and define how much the stimulus is coherent with each of the two bases. *n(t)* is the superimposed noise while a_0 is the offset.

When signal are sampled, the equation has to be written in a discrete form. In this case the signal is evaluated for each sample that belongs to the observed window obtaining the following matricial form:

$$S = AX + N$$

[Eq. 3.4]

When:

$$S = \begin{bmatrix} P(k) \\ P(k+1) \\ \vdots \\ P(k + window - 1) \end{bmatrix} \quad A = \begin{bmatrix} 1 & \cos 2\pi f t_1 & \sin 2\pi f t_1 \\ 1 & \cos 2\pi f t_2 & \sin 2\pi f t_2 \\ \vdots & \vdots & \vdots \\ 1 & \cos 2\pi f t_N & \sin 2\pi f t_N \end{bmatrix} \quad X = \begin{bmatrix} a_0(k + {}^{window}/_{2}) \\ a_1(k + {}^{window}/_{2}) \\ b_1(k + {}^{window}/_{2}) \end{bmatrix}$$

Where the X matrix contains the coefficients which need to be estimated for each sliding window ς.

The window is easily calculated, as the quotient between the sampling frequency (200Hz) and the frequency of the stimulus (5Hz). Thus the resulting width of the window is 40:

$$window = \frac{f\ sample}{f\ signal} = \frac{200\ Hz}{5\ Hz} = 40$$

The solution of the fitting problem is given by the next equation:

$$X = BS$$

[Eq. 3. 2]

B matrix is the pseudo-inverse:

$$B = (A^T * A)^{-1} * A^T$$

[Eq. 3. 3]

Fortunately – with respect to computational effort - B matrix needs to be computed just the first time, because it never changes in the different windows. The computation of X vector is actually performed both for the pressure (X_P) and flow (X_F) samples of corresponding window. [Eq. 3.4]] can be written even using Eulero's form:

$$S(t) = n(t) + a_0 + \Re[(a_1 + jb_1)e^{j2\pi ft}]$$

[Eq. 3. 4]

According to this, X_P and X_F are separated in the two components: *a* and *b*.
The impedance computation in the slot window ς becomes:

$$Z_{RS}(\varsigma) = \frac{P}{\dot{V}} = \frac{a_{P,1}(\varsigma) + jb_{P,1}(\varsigma)}{a_{\dot{V},1}(\varsigma) + jb_{\dot{V},1}(\varsigma)}$$

[Eq. 3. 5]

The resistance is the real part while the reactance is the imaginary part:

$$R_{RS}(\varsigma) = \Re(Z_{RS}(\varsigma))$$

[Eq. 3. 6]

$$X_{RS}(\varsigma) = \Im(Z_{RS}(\varsigma))$$

[Eq. 3. 7]

After each estimation ς, the window is shifted of one sample and the calculus is repeated.

This algorithm has been introduced in the Android program. The suited file named CPU.java executes the steps necessary to estimate R_{RS} and Z_{RS}. This part of the software does even the correction calculated and explained in the third chapter.

3.4 Pressure-Flow shift compensation

As said, the ventilator has been estimated to be good, but some changes has demonstrated necessary to study the respiratory mechanic of patients. Values of respiratory impedance obtained with by the ventilator, are compared to the other obtained through a gold standard setup. A correction in time shift has found in order to have exact time synchronization between flow and pressure data.

Flow and Pressure Correction – HFO mode

To find the correction, two setups are used - Figure 3. 9. As described in the down depicted figure, the ventilator exploits HFO mode in order to produce forced oscillations. In the left side of the figure, it is represented the gold standard setup, while in the right part, there is instead the developed setup, to be tested. The delivery of airflow waveform is thus the same, both in *side A* and *side B*, but the way of recording changes.

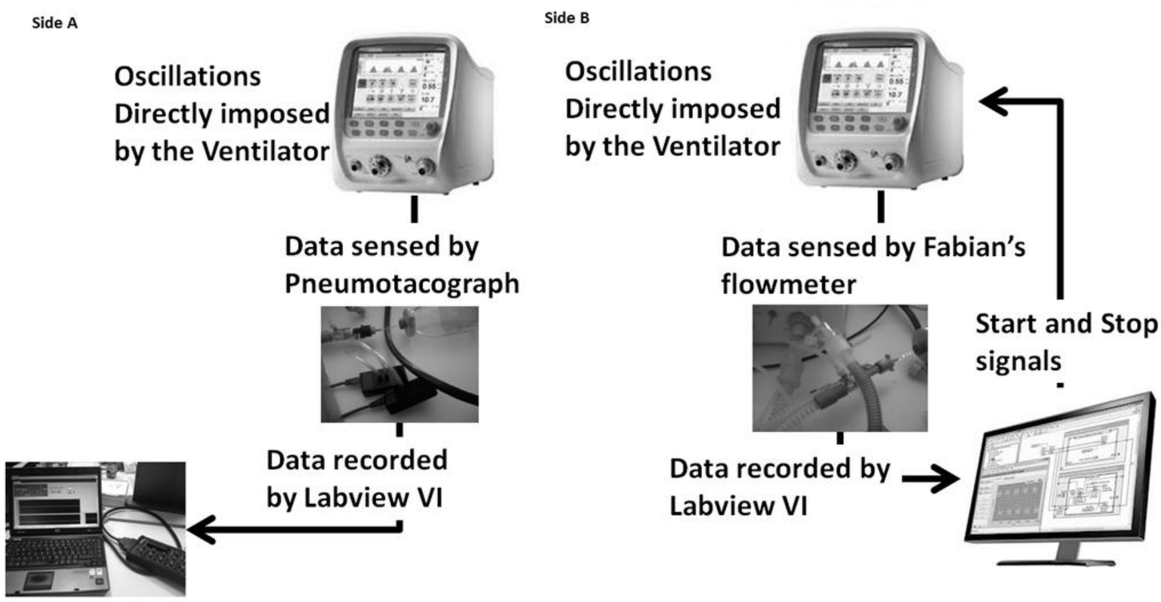


Figure 3. 9 HFO mode of ventilation - oscillations internally generated. Two acquisitions:

Side A: through National Instrument's DAQ Side B: through data transferred by ventilator

To find the correction we used a bottle and some glass capillaries, as lung test. The codename of the lung model is *TL1 R5*. All the models used will be described in detail in the next chapter. Acquisitions have been done using two personal computers, both using Labview software.

The PC – make reference to *side A* – records values of pressure and flow sensed by the gold standard sensor. In *setup B* instead, the tablet performs the record. The experiments have been done independently (side A and side B) and in different conditions, in order to sweep a lot of frequencies and to find out the right correction.

Tl1 R5 stands for "bottle #1 and 5 glass capillaries". Bottle #1 is used in series with 5 capillaries in parallel – consider Figure 3. 10.

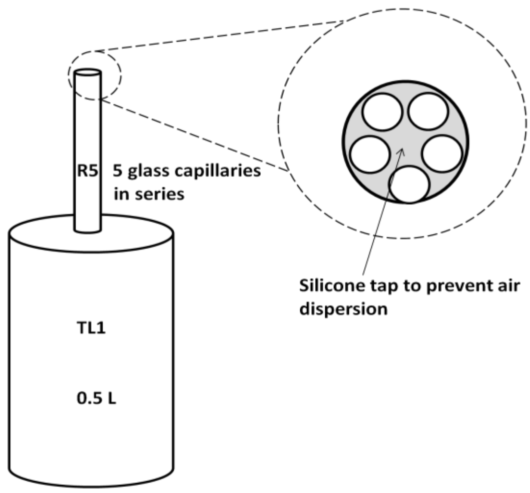


Figure 3. 10 Lung test with bottle#1 and R5 glass tube.

The following chart shows data obtained with the first set up, both from the gold standard, and from the ventilator. Values of respiratory impedance are calculated, in this first phase of investigation, offline through Matlab software. The rationale which permits the calculation is the same that is described in the Android application. In Tab 3. 1 it is thus possible to observe the mean values of resistance *R*, reactance *X* and the derived absolute *A* and phase *P*. The prefixes *Gold_std's* and *Fabian's* represent respectively the pneumotacograph, recorded thanks to a National Instrument Data Acquisition device (DAQ), considered as gold standard, and the Fabian ventilator data.

The recorded values are extremely different from the gold standard. This is the major evidence which confirms that flow and pressure data are not perfectly synchronized for small changes, such as the one requested by FOT - Tab 3. 1:

Tl1 R5	HFOV mode						
	Freq [Hz]	5	7	9	11	13	15
Gold Standard's	R [cmH2O*s/l]	64.39	65.87	67.31	68.59	69.16	69.58
Fabian's	R [cmH2O*s/l]	-14.92	-26.70	-40.37	-54.71	-68.96	-79.46
Gold Standard's	X [cmH2O*s/l]	-73.52	-49.22	-34.48	-24.20	-16.26	-9.64
Fabian's	X [cmH2O*s/l]	-91.09	-74.77	-62.82	-49.74	-32.81	-11.31
Gold Standard's	A [cmH2O*s/l]	97.74	82.23	75.62	72.74	71.05	70.25
Fabian's	A [cmH2O*s/l]	92.34	79.41	74.67	74.07	76.36	80.25
Gold Standard's	P [rad]	-0.85	-0.64	-0.47	-0.34	-0.23	-0.14
Fabian's	P [rad]	-1.73	-1.91	-2.14	-2.40	-2.70	-3.00
	Diff Phi [rad]	-0.88	-1.27	-1.67	-2.06	-2.47	-2.86

Tab 3. 1 Comparison between values sensed, and derived, by Fabian's sensors and by the gold standard's setup - HFO mode

The strange thing is that values of respiratory resistance appears to be negative, while it is physically impossible. The correction found, exploits a wide range of frequencies, from 5 Hz to 15 Hz because these ones are the frequencies generated by the ventilator. There is a delay between pressure and flow data. The delay depends also on post processing and filtering. Data elaboration and filters are different in the ventilator between CPAP and HFOV. In the next graph, it is represented the difference in phase at each frequency evaluated.

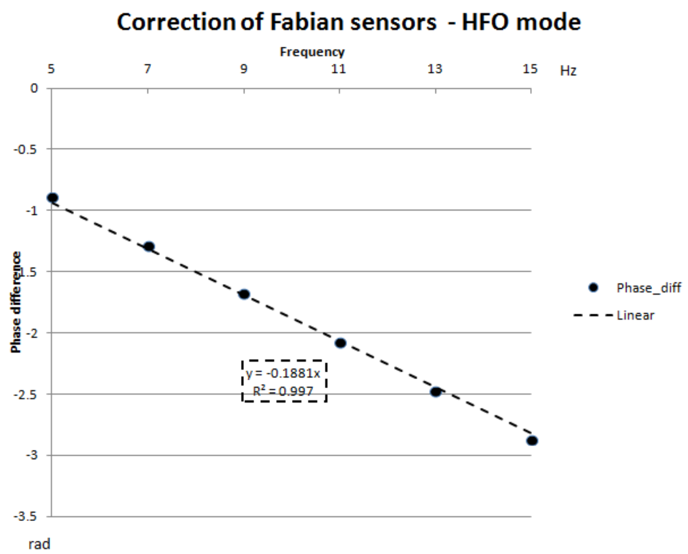


Chart 3. 1 Phase difference correction – phase difference(frequency) - HFO mode

Because the R^2 index is very good – $R^2 \geq 0.95$ - the regression is trustful. Thus the correction term is the slope of the regression:

$$\boldsymbol{M = -0.1881}$$

$$\boldsymbol{R^2 = 0.997}$$

The corrected values causes the following errors:

Tl1 R5 - HFOV

Freq (Hz)	5	7	9	11	13	15
Gold Standard R [cmH2O*s/l]	64.39	65.87	67.31	68.59	69.16	69.58
Error (R) [cmH2O*s/l]	6.4	2.8	0.1	2.2	2.3	1.9
Error (R)% [cmH2O*s/l]	9.1	4.1	0.2	3.3	3.5	2.8
Gold Standard X [cmH2O*s/l]	-73.52	-49.22	-34.48	-24.20	-16.26	-9.64
Error (X) [cmH2O*s/l]	1.7	2.5	2.1	1.1	1	2.6
Error (X)% [cmH2O*s/l]	2.3	5.4	6.5	4.7	5.6	21.1

Tab 3. 2 Error between resistance and reactance values, calculated by Fabian's sensors_corrected and by the gold standard's setup. Error is displayed both as absolute, and relative – in percentage. Values belonging to Gold standard's setup are displayed too - HFO mode

In Tab 3. 2 values differs in percentage less than the 10%. When values are very close to zero - as it happens in reactance at high frequencies – is not so reliable to observe error in percentage. Make reference to X@15 Hz: a difference of 21% seems not to be good, but the absolute difference is less than 3.

Flow and Pressure Correction – CPAP mode

When other mode of ventilation are used, such as CPAP, it is not possible to use the correction factor obtained before. In fact as said, there are different process and filtrage of data. FOT can be used even with this mode of ventilation, with oscillations externally applied. The setup is down depicted - Figure 3. 10 :

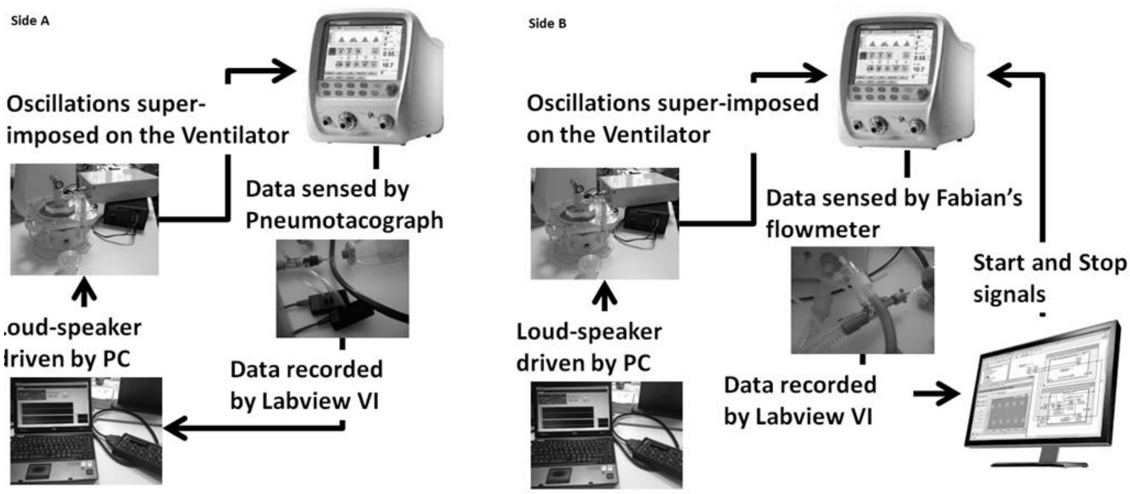


Figure 3. 11 CPAP mode of ventilation with oscillations externally generated. Two acquisitions:

Side A: National Instrument's DAQ with standard sensors Side B: data transferred by ventilator

Acquisitions have been done again using two personal computers, both using Labview software. The PC – make reference to *side A* - drives the loudspeaker to super-impose high frequency oscillations. In *setup B* instead, two pc are used; the first is the one who drives oscillations, while the second, in the right part of *side B*, behaves exactly as the Android application does, starting and stopping the records, and saving data. Pressure and flow info, are sensed by pneumotacograph in the first image, and by flow-meter and pressure-meter in the second case.

The following chart shows data obtained with the first set up, both from the gold standard, and from the ventilator. In Tab 3. 3 Comparison between values sensed, and derived, by Fabian's sensors and by the gold standard's setup it is thus possible to observe the mean values of resistance *R*, inertance *X* and the derived absolute *A* and phase *P*. The prefixes *Gold_std's* and *Fabian's* represent respectively the pneumotacograph, recorded thanks to a National Instrument Data Acquisition device (DAQ), considered as gold standard, and the Fabian ventilator data. To be noticed that in this second setup, exploiting external oscillations, a wider range of frequencies have been investigated. In this table all these frequencies are displayed, but the correction has been researched only in some values – the same used in the previous paragraph.

Tl1 R5	**CPAP mode**						
	Freq [Hz]	5	7	9	11	13	15
Gold Standard's	R [cmH2O*s/l]	67.10	68.67	68.56	68.77	70.81	71.93
Fabian's	R [cmH2O*s/l]	59.14	58.71	57.09	60.25	59.71	61.77
Gold Standard's	X [cmH2O*s/l]	-71.39	-47.36	-33.47	-23.75	-16.68	-10.14
Fabian's	X [cmH2O*s/l]	-71.58	-51.79	-39.94	-35.64	-30.70	-28.92
Gold Standard's	A [cmH2O*s/l]	97.98	83.41	76.29	72.75	72.89	72.66
Fabian's	A [cmH2O*s/l]	92.85	78.32	69.68	70.00	67.14	68.20
Gold Standard's	P [rad]	-0.82	-0.60	-0.45	-0.33	-0.23	-0.14
Fabian's	P [rad]	-0.88	-0.72	-0.61	-0.53	-0.47	-0.44
	Diff Phi [rad]	-0.06	-0.12	-0.16	-0.20	-0.24	-0.30

Tab 3. 3 Comparison between values sensed, and derived, by Fabian's sensors and by the gold standard's setup - CPAP mode

While the module values are very similar, we obtained a difference in the phase with a linear trend with frequency. In Chart 3. 2, the plotted differences are the starting points of a linear regression, Because the R^2 index is very good – $R^2 \geq 0.95$ - the regression is trustful. Thus the correction term is the slope of the regression:

M= -0.00187

R^2= 0.9761

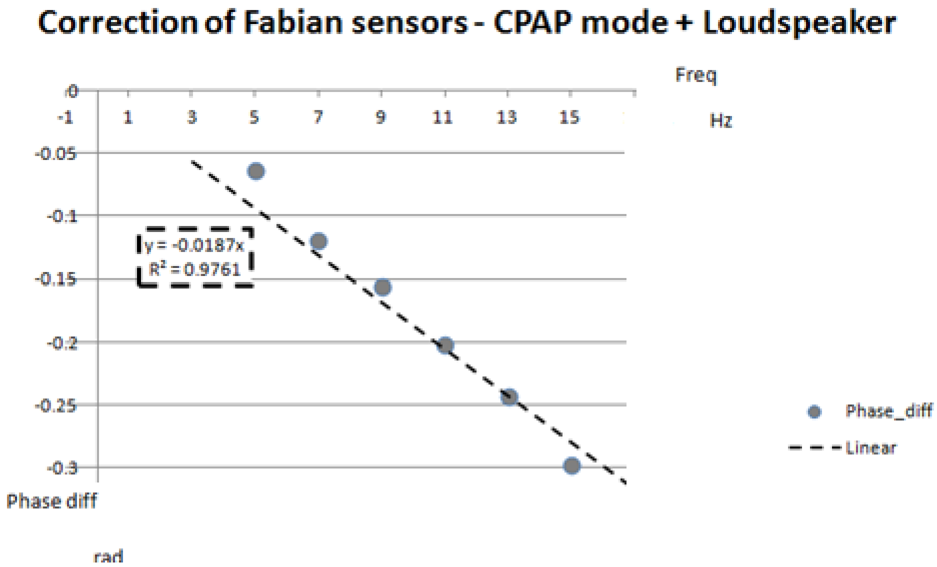


Chart 3. 2 Phase difference correction – phase difference(frequency) - CPAP mode

This behavior is compatible to time shift, or delay, between pressure and flow data, as explained before.

Correcting for this time-shift the *Fab* data, the remained difference with the gold standard values are (Tab 3. 4):

Tl1 R5 - CPAP

Freq (Hz)	5	7	9	11	13	15
Gold Standard R [cmH2O*s/l]	67.09757	68.67379	68.56247	68.76666	70.81185	71.92766
Error (R) [cmH2O*s/l]	2.7	5	1.5	1.8	21	3.3
Error (R)% [cmH2O*s/l]	3.9	0.8	2.2	2.5	3	4.8
Gold Standard X [cmH2O*s/l]	-71.3901	-47.3611	-33.4677	-23.745	-16.6789	-10.137
Error (X) [cmH2O*s/l]	1.4	0.8	1.7	0.3	0.5	0.7
Error (X)% [cmH2O*s/l]	2	1.7	5.4	1.2	2.8	6.9

Tab 3. 4 Error between resistance and reactance values, calculated by Fabian's sensors_corrected and by the gold standard's setup. Error is displayed both as absolute, and relative – in percentage. Values belonging to Gold standard's setup are displayed too - CPAP mode

It can be easily observed the difference in percentage between the gold standard values and the corrected values. Differences in absolute values are lower than 5 cmH2O*s/l for R and 2.0 for X.

4 - IN VITRO MEASURES

Once the measurement system for the lung mechanics monitoring (described in the previous chapter) had been developed, its accuracy was tested *in vitro*.

The *in vitro* tests presented in this chapter were designed to address the following issues:

- verify the accuracy of Z computation after the application of the computed compensation of time shifts between pressure and flow samples;
- verify the possibility of applying FOT during SLI;
- test the performances of the tablet in terms of missing data and data processing.

4.1 Lung Models

Since the respiratory system is very complex, it may be useful to consider it as an interconnection of three simple passive elements - as described in the first chapter - each one corresponding to a different mode of energy handling: storage by means of elastic potential energy, storage by means of kinetic energy and dissipation by means of friction.

All measurements were performed on a mechanical analogue of the infants respiratory system. The respiratory system can thus be represented by an R-I-C series combination.

In the electric analogue of the respiratory system reported in Figure 4. 1, viscous resistances and inertive properties of the airways are represented by a resistor (R_{AW}) and by an inductor (I_{AW}) respectively, while a capacitor (C_L) simulates the compliance of the respiratory system.

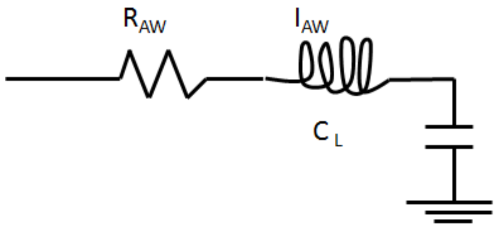


Figure 4. 1 R-I-C Lung model

To represent the infant respiratory system, glass tubes and rigid bottles are combined in parallel or in serial to obtained resistance and reactance values comparable with resistance and reactance of the infants' respiratory system obtained by previous measurements (Chapter 3 – Paragraphs 3.4). Let consider the elements in Figure 4. 1 former stands for resistance and inertance, while the latter represents lung compliance, referring to a volume assumed as adiabatic compressible; air is considered a biatomical gas. To be noticed that compliance is the inverse of *elastance*.

A suited Matlab function has been used to calculate the parameters. The physical basis are represented in the schema below:

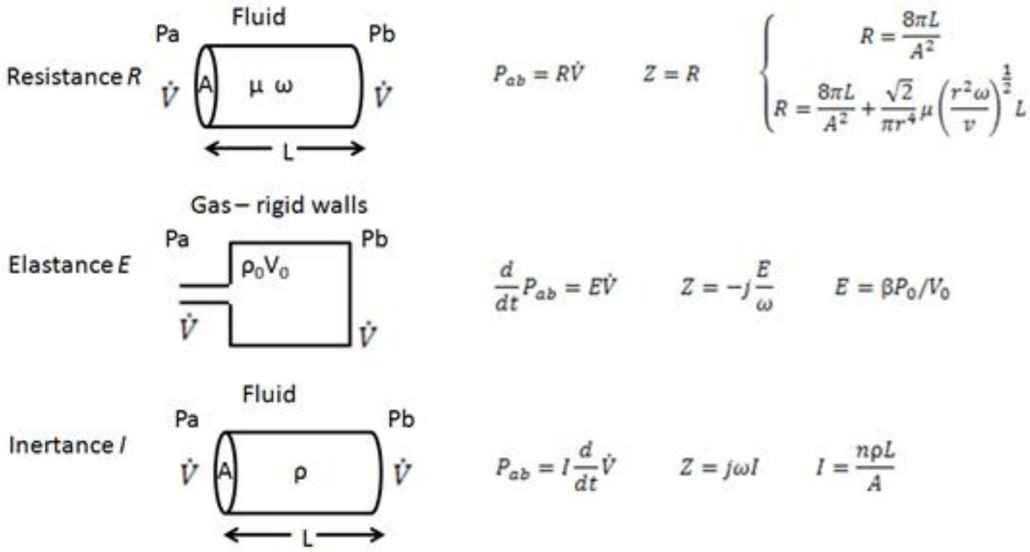


Figure 4. 2 Resistance, Elastance and Inertance manners of calculation

Using mainly two glass bottles and 3 different glass tubes, 3 lung models have been created. The Lung model components are:

Tl1 = glass bottle with 0.5 L capacity

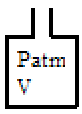
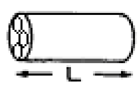
Tl2 = glass bottle with 0.7 L capacity

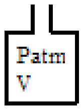
R1 = glass catheter

R5 = glass catheter with 5 capillaries in parallel

R6 = glass catheter with 6 capillaries in parallel

As written, *R* represents the resistance given by the glass capillaries. The number stands for the number of capillaries in parallel. *TL* are the bottle #1 and #2 which have different geometrical capacity. In the Tab 4. 1 below the theoretical values are represented:

Lung Test	Compliance	Resistance	Inertance
Tl1 R5	Rigid bottle: 	Flow resistor realized by n capillaries arranged in parallel: 	
	V = 0.5 l	L = 10 cm ID = 1.25 mm	n = 5
	0.35 mL/cmH_2O	60.92 cmH_2O*s/L	0.11 cmH_2O*s^2/L

Tl2 R1	Rigid bottle:	Flow resistor realized by n capillaries arranged in parallel:		
				
	V = 0.7 l	L = 10 cm		n = 1
	0.69 mL/cmH_2O	11.43 cmH_2O*s/L		0.0004 cmH_2O*s^2/L
Tl1 R6	Rigid bottle:	Flow resistor realized by n capillaries arranged in parallel:		
				
	V = 0.5 l	L = 10 cm	ID = 1.25 mm	n = 6
	0.35 mL/cmH_2O	50.77 cmH_2O*s/L		0.11 cmH_2O*s^2/L

Tab 4. 1 Lung Test parameters calculation

Once again, the lecturer makes reference that compliance *C* is the inverse of elastance *E*.

4.2 Setup Validation

To validate the setup for the measurement of oscillatory mechanics, different tests have been performed.

- Firstly, we focused the attention on the correction of the time shift between pressure and flow and tested it on the different lung models presented in the previous paragraph. We used different lung models in order to verify that the correction applied for Z computation is not dependent on the mechanical characteristics of the test lung used to identify the correction coefficients. This is fundamental to obtain an accurate measurement of *Z* over the whole range of interest.
- After that, the possibility of applying FOT during SLI was tested
- Lastly, we checked the performances of the tablet both in terms of data transmission (we checked missing data) and of data processing.

As the ventilator can be used both in CPAP and HFO modality, depending on the way of generating the pressure stimulus, and as the time lag between pressure and flow is different in the two modalities, the implemented set up was validated using both CPAP and HFOV modes. In the next paragraphs, these tests will be described. Each one of the points described above, will be split in CPAP and HFO.

4.2.1 Z Computation

Validation in CPAP mode

As in the identification of a correction factor, two setups were used (see Figure 3. 11). One is the developed setup, and the other is the gold standard, used to verify the goodness of correction. Bottle #1 in series with 5 capillaries arranged in parallel was used to identify the optimal time shift correction factor between flow and pressure data.
To validate the correction factor, it was tested with two other test lungs: Tl2R1 and Tl1R6. The errors in Z estimation for the two model are reported in Tables 1 and 2 respectively. The tables include even the resistance and reactance values sensed by the gold standard setup. It has been decided to add also these values, to permit a better observation of data.
Errors were calculated as follows:

Error = Absolute Error: *Abs(data1 – data2)*

Error % = Relative Error in percentage: *Error/ data1 *100*

Tl2R1

Freq (Hz)	5	7	9	11	13	15
Gold Standard R [cmH2O*s/l]	8.76	8.33	8.19	8.33	8.35	8.57
Error (R) [cmH2O*s/l]	0.03	0.37	0.44	0.83	1.16	0.59
Error (R)% [cmH2O*s/l]	0.38	4.37	5.28	9.91	13.76	6.86
Gold Standard X [cmH2O*s/l]	-39.03	-27.45	-20.86	-16.41	-13.33	-10.86
Error (X) [cmH2O*s/l]	1.15	0.38	1.12	0.20	0.72	0.11
Error (X)% [cmH2O*s/l]	3.08	1.45	5.61	1.28	5.67	1.09

Tl1R6

Freq (Hz)	5	7	9	11	13	15
Gold Standard R [cmH2O*s/l]	54.72	55.88	57.14	58.30	59.54	59.82
Error (R) [cmH2O*s/l]	5.05	3.07	1.74	1.44	0.91	1.08
Error (R)% [cmH2O*s/l]	9.15	5.54	3.12	2.54	1.57	1.83
Gold Standard X [cmH2O*s/l]	-94.23	-64.38	-46.54	-34.31	-25.24	-17.83
Error (X) [cmH2O*s/l]	1.80	0.86	0.84	1.70	1.71	1.91
Error (X)% [cmH2O*s/l]	2.00	1.43	1.92	5.39	7.30	11.37

Tab 4. 2 Error between resistance and reactance values, calculated by Fabian's sensors_corrected and by the gold standard's setup. Error is displayed both as absolute, and relative – in percentage. Values belonging to Gold standard's setup are displayed too. The two validation lung tests are used.

To be compliant with European FOT guidelines the percentage error should be below 10%. It is possible to observe that the percentage error is always below 10% but in two cases: in *TL2R1* at 13 Hz (resistance) and in *Tl1R6* at 15 Hz (reactance).

However the values exceeding this threshold correspond to low absolute errors: 1.16 cmH2O*s/l and 1.91 cmH2O*s/l respectively.

The agreement between the Zrs measurements performed with our set-up and with the gold standard was also tested using the Bland-Altman analysis.

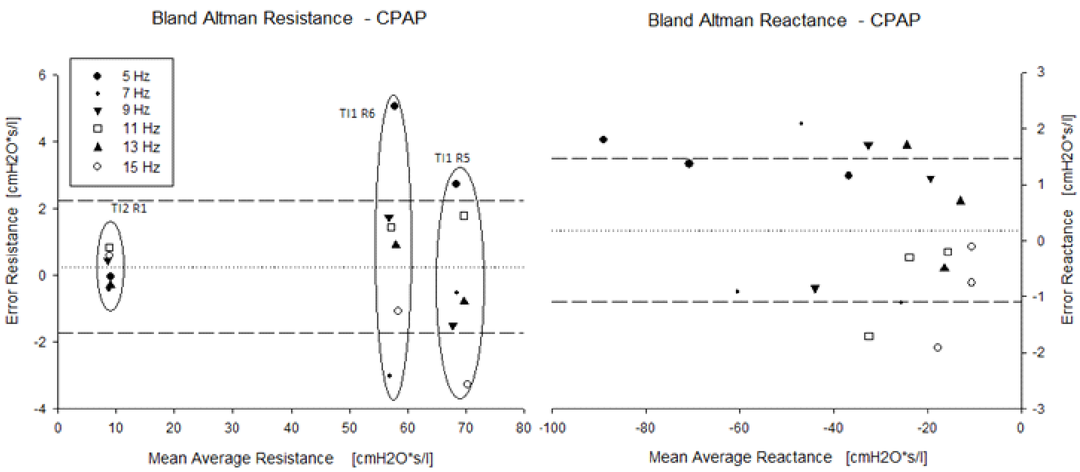
Briefly, Bland-Altman analysis is a graphical statistical method to compare two measurement techniques.

The x axis represents the mean value of the two measurements, the y axis represents the absolute error:

$$Y\ axis:\ \ data_1 - data_{2;}$$

$$X\ axis:\ \ (data_1 + data_2)\ /2$$

Bland and Altman plots were extensively used to evaluate the agreement among two different instruments or two measurements techniques. Bland and Altman plots allow us to investigate the existence of any systematic difference between the measurements (i.e., fixed bias), to identify possible outlier and to see whether there is any trend between the differences and the mean values better than with linear regression., In the graph the mean absolute error is reported – the middle horizontal line – together with *mean* ± *2* standard deviation*. To be noticed that values can be outside the external lines – 95% of samples are comprised between the two lines – anyway it is important to consider the numeric distance between them (limit of agreement). To be noticed Graph 4. 1



Graph 4. 1 Bland Altman Graph to consider validation tests. Left: Resistance. Right: Reactance. On the x axis the mean value between gold standard and developed setup's values; on the y axis the error between them - CPAP mode

The graphs depicted above refer to the two test lungs used to verify that the correction applied for Z computation is not dependent on the mechanical parameters of the test lung used for correction computation. The two components of respiratory impedance are displayed separately: the left panel represents the real component – resistance - while the right panel represent the imaginary component – reactance. In the resistance plot, it is easy to distinguish the three test lungs.

Based on the previous tables and on these graphs we can draw the following conclusions:

- for reactance measurements there is no trend between the absolute error and the mean value
- the absolute error in resistance measurement is higher for higher resistance values. This is expected: with higher resistance, the flow signal is lower at the same pressure stimulus and can reach the anemometer resolution limits. Anyways the percentage error remains below 10%.

Validation in HFO mode

When this modality is used pressure oscillation are generated using the HFO module of the ventilator (see Figure 3. 9 in the previous chapter). The setups used to verify the goodness of correction, are the developed one, and the gold standard one.

The validation was performed using the same test lungs described above. AS done before, values of resistance and reactance, sensed by the gold standard setup, are added. This, to permit a better observation of data - Tab 4. 3:

Tl2R1

Freq (Hz)	5	7	9	11	13	15
Gold Standard R [cmH2O*s/l]	9.05	8.56	8.35	8.38	8.45	8.55
Error (R) [cmH2O*s/l]	1.33	0.94	0.76	0.23	0.20	0.56
Error (R)% [cmH2O*s/l]	15.18	11.26	9.24	2.76	2.39	6.50
Gold Standard X [cmH2O*s/l]	-37.36	-25.91	-18.89	-15.53	-12.64	-10.48
Error (X) [cmH2O*s/l]	0.79	1.16	0.98	0.92	0.24	0.18
Error (X)% [cmH2O*s/l]	2.01	4.24	4.72	5.61	1.81	1.69

Tl1R6

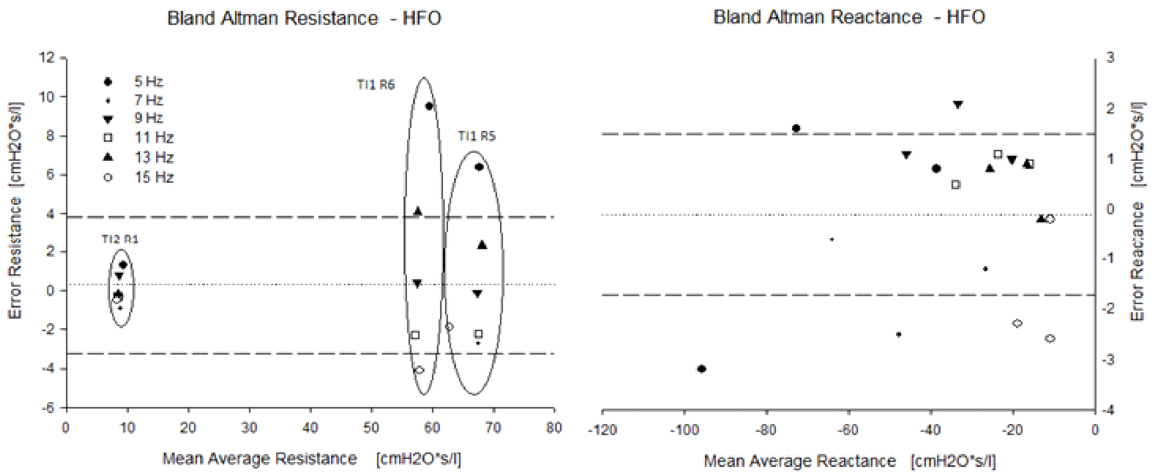
Freq (Hz)	5	7	9	11	13	15
Gold Standard R [cmH2O*s/l]	5524	55.33	55.94	56.38	57.57	58.89
Error (R) [cmH2O*s/l]	9.60	4.18	0.41	2.33	4.08	4.06
Error (R)% [cmH2O*s/l]	17.53	7.48	0.72	3.99	6.85	6.78
Gold Standard X [cmH2O*s/l]	-89.85	-60.88	-43.54	-31.63	-23.47	-16.80
Error (X) [cmH2O*s/l]	3.20	0.63	1.05	0.51	0.77	2.30
Error (X)% [cmH2O*s/l]	3.40	0.97	2.26	1.47	3.04	12.91

Tab 4. 3 Error between resistance and reactance values, calculated by Fabian's sensors_corrected and by the gold standard's setup. Error is displayed both as absolute, and relative – in percentage. Values belonging to Gold standard's setup are displayed too. The two validation lung tests are used.

Percentage Errors are almost always lower than 10% - acceptable based on FOT guidelines. In Tl2R1 there are two values of resistance with a percentage error higher than 10%. Anyway the absolute error is very low: under 1.5 cmH2O*s/l.

In Tl1R6 there is a similar situation at high frequency (15Hz): reactance exceeds the 10% error, but the absolute error is lower than 2.5 cmH2O*s/l. Only one value is much greater than 10% and its absolute error is 9.6 cmH2O*s/l.

Graph 4. 2 shows the Bland-Altman analysis between R_{RS} and X_{RS} measurements performed using the gold standard set-up and Fabian in HFO modality.



Graph 4. 2 Bland Altman Graph to consider validation tests. Left: Resistance. Right: Reactance. On the x axis the mean value between gold standard and developed setup's values; on the y axis the error between them - HFO mode

As in CPAP mode, also in HFO mode there is no trend in the reactance graph, while the absolute error in resistance is higher for higher resistance values. However the percentage errors remains below 10%.

In conclusion, both CPAP and HFO modalities allow the accurate measurement of resistance and reactance in the range of interest.

4.2.2 FOT Measurement During SLI

Once established that it is possible to obtain accurate FOT measures using Fabian, we evaluated which is the best way to study lung mechanics during a SLI.

In order to perform the sustained lung inflation (SLI) we decided to connect the ventilator circuit to a manual T-piece resuscitation circuit. Such circuit consists in a tube connected to a gas flow source and ending with a positive end-expiratory pressure (PEEP) cap. As explained in the previous chapter, when the cap is left opened the pressure delivered to the infant is equal to the set PEEP value. When the PEEP cap is manually closed, the pressure suddenly raises to a pre-set positive inspiratory pressure (PIP) value.

In order to have a quick pressure rise during SLI it is important that also the ventilator and its circuit respond quickly to the pressure change.

To understand the time response of the ventilator during the two modalities, a simple test was performed: the two setups were connected to the T-piece resuscitation circuit. The PEEP cap was alternatively closed and opened in order to simulate SLI maneuvers.

The figure reported below represents a SLI manuevre during HFO mode ventiation. The PIP is set to 25 cmH_2O but the PEEP cap permits an expiratory pressure of 5 cmH_2O – to be noticed that 1 cmH2O = 0.98 mbar, the measure unit depicted - Figure 4. 3. Oscillations at 5 Hz are superimposed, with an amplitude of 5 cmH_2O peak-to-peak:

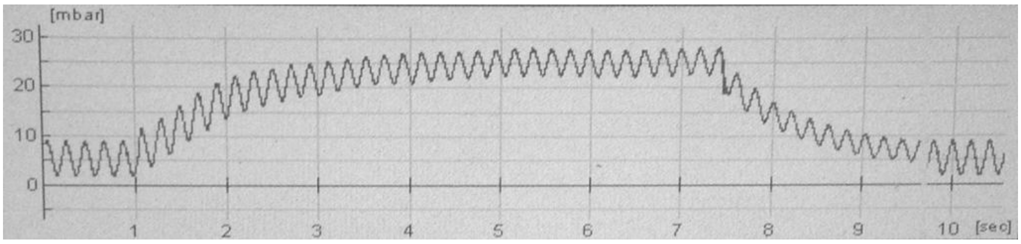


Figure 4. 3 SLI performance with oscillations internally implemented - HFO mode

It is easy to observe that the pressure rise is quite slow, pressure takes about 6 seconds to stabilize to value of 25 cmH_2O.

Figure 4. 4 represents the same trial during in CPAP mode. The PEEP and PIP values the same: 5 and 25 cmH_2O respectively (equivalent to approximately 5÷25 mbar).

Figure 4. 4 SLI performance using CPAP mode

When the user manually closes the PEEP cap, the PIP value is reached very quickly. When the cap is released, the value goes back to 5 cmH_2O in less than 1 s. This setup presented a very good time response.

Anyway, there are some disadvantages. Because of the very fast response, the PIP pressure overcomes the pre-set value for a while of about 5 cmH_2O. This could be a problem since too high pressures can produce pneumotoraces.

Moreover, a setup suitable for the NICU and the delivery room should be as compact as possible. For this reason, the use of an external oscillations generatoris not satisfactory.

This is one of the reasons why we preferred to use the HFO modality.

4.2.3 Data Communication and Impedance Computation Test

The tests described below were performed to test the performances of the Android application. In particular we were interested in verifying that there are no lost samples and that the implemented algorithm for Z_{RS} computation works properly. In the previous tests, data were elaborated offline, using Matlab software after the record of flow and pressure. Instead the application calculates directly R and X, on line.

Data MissingCheck

To verify that no data went missing, several long acquisitions has been done with the ventilator sensors and the gold standard sensors in series. In this way, it is possible to record the same tracks. To be compared, tracks need to be synchronized as registration of the signals didn’t start at the same moment. Once realigned by cross-correlation, data are verified to have the equal number and it is thus demonstrated that no data get missed. The test is performed off line, using Matlab software again.

Figure 4. 5 is representative that no data get lost during the recording. Only the pressure track is displayed.

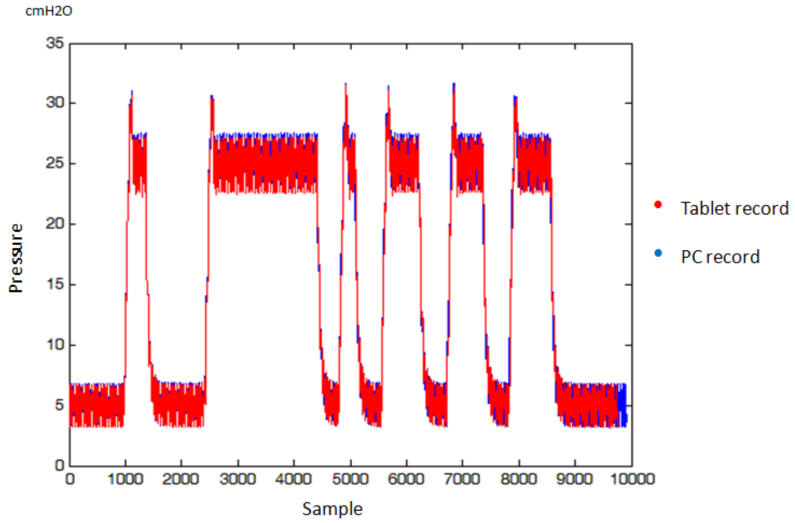


Figure 4. 5 Serial measures in comparison - synchronized

The acquisition using tablet proved to work properly.

Data Elaboration Check

To verify whether the Z computation works properly, it is necessary to compare measurements performed on a test lung calculation using two kinds of device - PC Virtual Instrument and Tablet application. This last test exploits a single record. Data are recorded by the Tabet and Z_{RS} is calculated and corrected in real time. Pressure and Flow, data are also saved and processed offline. In the next figures, pressure, Resistance and Flow data are showed. It is easy to understand that Z_{RS} corrected is the same both in Offline and Runtime correction.

Values of resistance are showed, both in offline and online analysis. Pressure waveform is showed in order to verify that during expiration and inspiration, the correspondent resistance does not change - Figure 4. 6. Transitory moments are not to be considered because of low signal to noise ratio.

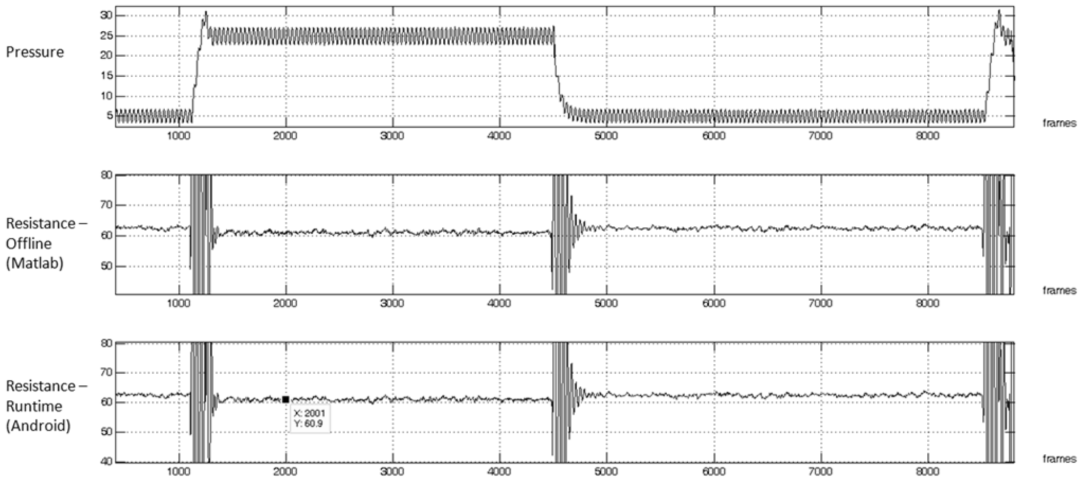


Figure 4. 6 Comparison between online and offline calculated resistance

The previous considerations can be done observing the reactance values – make reference to Figure 4. 7

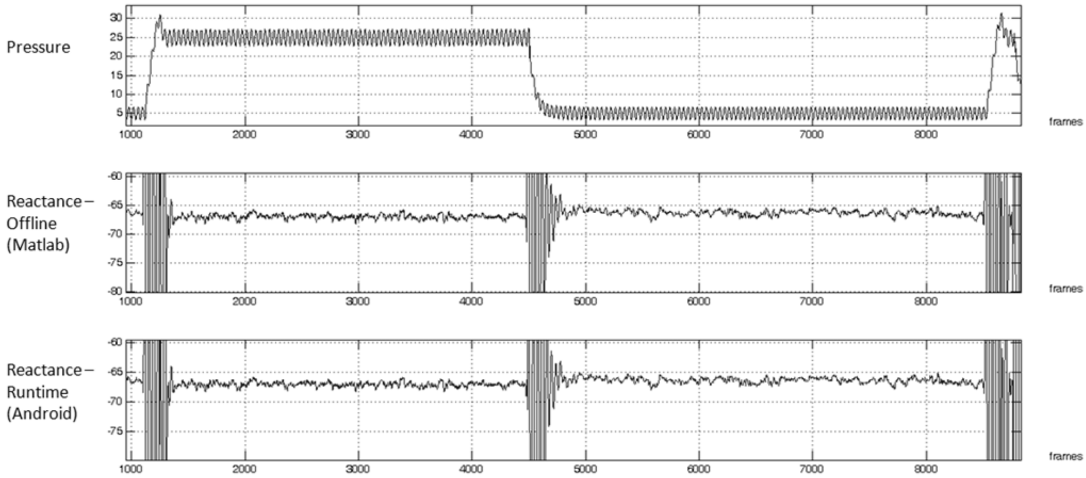


Figure 4. 7 Comparison between online and offline calculated reactance

To definitely validate the correction, even using the Android application, the stable tracts are chosen. Mean and standard deviation values are calculated and showed in Tab 4. 4:

[cmH2O*s/l]	R_android	R_matlab	X_android	X_matlab
Mean	62.03	620.3	-66.61	-66.61
Std dev	0.96	0.95	0.81	0.77

Tab 4. 4 Representation of values of the same track, offline and online calculated

R and *X* prefixes refer to *resistance* and *reactance*. The suffixes *android* and *matlab* refer respectively to the online and offline analysis. Measurements done in runtime and offline are quite the same.

Thus this last test definitely ensures the correctness of the application.

5 – IN VIVO MEASUREMENTS

After the in vitro tests, the developed set up was tested in vivo on 3 preterm newborns. The values of resistance and reactance were compared with the ones obtained by another already validated - but cumbersome - set up. SLI was then performed on these three subjects to verify the clinical feasibility of these measurements. The results of all this are reported in this chapter.

5.1 Set-up

The mechanical analogue of the infant respiratory system exploited a rigid bottle and a bunch of glass capillaries. This is a linear system that can be tested in ideal conditions. When it comes to newborns, respiratory system linearity and ideal conditions does not subsist anymore. Therefore a feasibility study and an *in vivo* validation are highly recommended before using the measurement set-up for other studies. It is important to verify the reliability of the set-up when measurements are performed in the clinical environments, which presents several criticalities that may limit the accuracy of the data.

In order to test the accuracy of the set-up measurements were alternatively performed using the new set-up and a gold standard. As a gold standard we used an already validated set-up, based on a servo-controlled linear motor able to superimpose oscillations to the CPAP applied by the ventilator.

The clinical validation setup, reported below in Figure 5. 1, is very similar to the one which exploited the loudspeaker. Anyway because of space reason and because of the place where records have been done – nursery hospital room – this already tested device has been preferred.

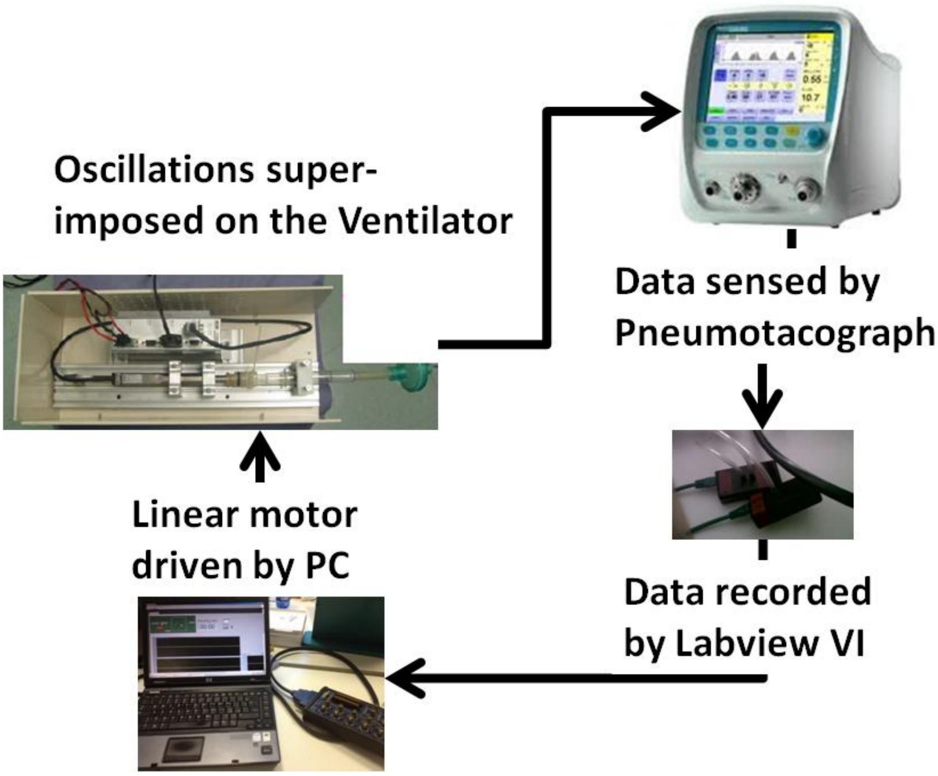


Figure 5. 1 Clinical validation setup: oscillations are externally imposed by a PC driven Linear Motor

In both set-ups the positive pressure is delivered by the ventilator. The difference is that in the gold standard set-up the oscillations, required for the assessment of respiratory impedance using the Forced Oscillations Technique, are generated by an external linear motor, driven by a PC. Data are recorded by the PC itself and consequently calibrated, both in pressure and flow data.
Respiratory impedance is calculated off-line, like in the set-up described in the previous chapter. The linear motor is a Linmot modular built servo drive system. Its movements are controlled in closed-loop by a PID (Proportional, Integrative, Derivative control) controller.

The measurements are performed independently, with the developed setup and using the above described setup.
The validation protocol consists in performing two independent measurements for each patient and comparing them. If the values are similar, the developed setup is considered reliable.

5.2 Protocol

This part of the study has been conducted in collaboration with *Clinica Mangiagalli di Milano* – in Milan. The protocol applied is very similar to the one described in the first chapter:
a mean PEEP of about 5 cm of water, through a facial mask is applied. Forced oscillations are superimposed. PIP is instead of 20 cm of water. The high peak of pressure is maintained for 15 seconds, after that the initial value of 5 cmH_2O is restored. When the FABIAN-based setup is used the mean pressure and the oscillations are provided by the using the ventilator in HFO mode; when the validation setup is used, the ventilator works in CPAP mode and oscillations are provided by the linear motor.
The SLI maneuver is performed by the attending physician by manually closing the PEEP can of the T-piece circuit. This maneuver rises the airway PIP to 20 cmH_2O.

5.3 Study Population

To test the feasibility and reliability of monitoring lung mechanics during SLI by this set up three preterm babies were recruited. While SLI is usually performed at birth, we studied older babies because we wanted to compare two measurements obtained using different set-ups, which is not possible at birth as the lung is changing quickly and measurements at different times may not reflect the same situation.

The characteristics of the studied population are reported in the following Tab 5. 1.

Patient	Gestational Age	Age	Weight at Birth
#1	30 weeks	4 weeks	1300g
#2	30 weeks	4 weeks	1000g
#3	34 weeks	1 week	1285g

Tab 5. 1 Table representing patients data

5.4 Data Processing

Data are sensed both by clinical setup and by the gold standard clinical setup.

When the developed set-up is used, the calculation of Z_{RS} is performed in real time by the application running on the tablet. When the validation setup is used data processing is performed off line. The algorithm used - based on LMS method - is the same. Every record is analyzed in terms of pressure, flow, resistance and reactance. The values chosen are those which have an average flow close to zero, meaning that the leaks were negligible. In these non ideal conditions, with newborns, it is difficult to select zero flow tracts because of the no collaboration of infants in wearing the facial mask, because of their continuous crying and, simply, because usually their breathe is superimposed to forced oscillations. When these criticalities occur, real and imaginary components of respiratory impedance inherit spikes which bring to not reliable values. It is thus fundamental to observe all the tracks avoiding spikes and leakages.

Anyway, the technician has to observe all the 4 tracks in a single view and evaluate which values choose. In the next paragraph, the next Figure 5. 2 will explain the rationale.

The discernment between satisfactory and not satisfactory tracts is manually performed. A further development will surely be the automatic recognition.

5.5 Results and Discussion

5.5.1 Z Comparison

The first part of the record, before SLI, represents the baseline respiratory mechanics without the result of any attempt to recruitment. Baseline condition has been used to test if the developed monitoring system is reliable also *in vivo*. The comparison between impedance values measured with the developed set-up and with gold standard is reported in the following Tab 5. 2:

	R [cmH2O*s/l]		X [cmH2O*s/l]	
Newborn #1	mean	std dev	mean	std dev
Gold Standard	73.7	10.9	-47.4	11.4
Clinical Setup	81.3	5.2	-60	6.8
	R [cmH2O*s/l]		**X [cmH2O*s/l]**	
Newborn #2	mean	std dev	mean	std dev
Gold Standard	94.3	9.1	-72.7	7.9
Clinical Setup	69.2	6	-62.8	6.8
	R [cmH2O*s/l]		**X [cmH2O*s/l]**	
Newborn #3	mean	std dev	mean	std dev
Gold Standard	31.9	4.1	-57	4.2
Clinical Setup	35.2	7.2	-48.7	4.1

Tab 5. 2 Resistance and Reactance values - expressed in mean and standard deviation – sensed by developed setup and gold standard's one.

The values are markedly different, especially resistance values in newborn # 2. However it is very difficult to compare two measurements performed *in vivo* in different moments because the neonatal lung is very unstable. Newborn # 3 is the one with the best match between values measured using the two set-up, likely because he was the most compliant to the measurements.

5.5.2 Lung Mechanics During SLI

After having tested the reliability of the developed setup, SLI maneuvers were performed on patients, using only the developed setup.

Figure 5. 2 below represents a whole record during a SLI maneuver. Four tracks are displayed: from top to bottom pressure, flow, resistance and reactance.

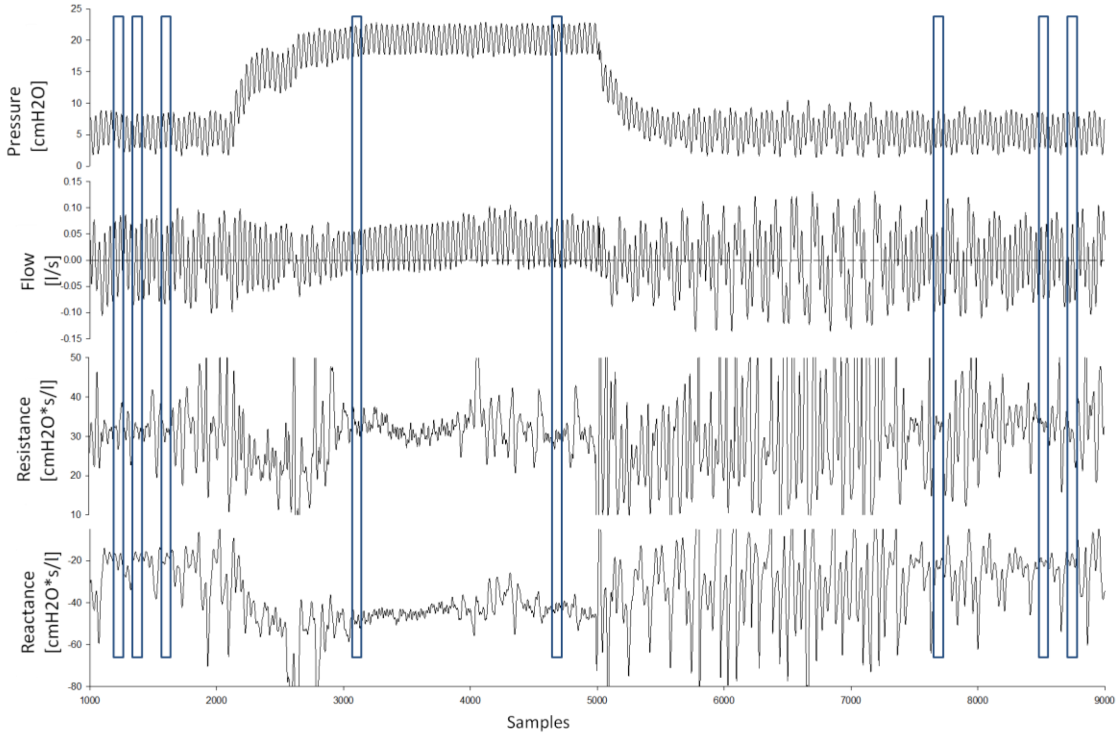


Figure 5. 2 Representation of a whole SLI approach, performed on a patient. there are four tracks represented. Blue boxes underlines the tracts retained to be of interest

On flow and pressure traces it is possible to identify both the sinusoidal FOT stimulus and the infants breathing signal, with higher and tight parts corresponding to inspiration and lower and wider ones corresponding to expiration.

On the pressure trace the SLI maneuver is clearly visible. When pressure was raised to 20 cmH_2O the infant reduced his breathing frequency dramatically. In fact in infants the Hering–Breuer inflation reflex plays an important role. Briefly when the lung tissue is stretched by inflation, the stretch receptors respond by sending impulses to the respiratory center, which in turn slows down the rate of inhalations.

When the pressure decreases again he hyperventilates. This could be due to the increase in CO_2 concentration due to ipo-ventilation during SLI.

During the maneuver the average of the flow signal slightly moves from zero because the effects of small leakages is amplified by the higher pressure difference between the inside and the outside of the facial mask. However, when the leakages are small, they represent a high impedance pathway in parallel to the infants’ lung impedance for 5Hz oscillation and therefore they do not affect the measurements significantly.

As the reactance is a function of lung volume, when infants breath the reactance value is considered at end-expiration in order to account for the number of alveoli that remain open during all the respiratory cycle and exclude the contribution of tidal recruitment.

When the infants breathing frequency increases, as immediately after SLI, it becomes difficult to identify end expiration with reliable Z values as the frequency components of the breathing signal and the FOT stimulus become overlapped. For this reason post-SLI Z was computed several breath after SLI.

Blue boxes enclose three end expiratory tracts before SLI, three after SLI and two tracts at the beginning and at the end of SLI with reliable Z computation.

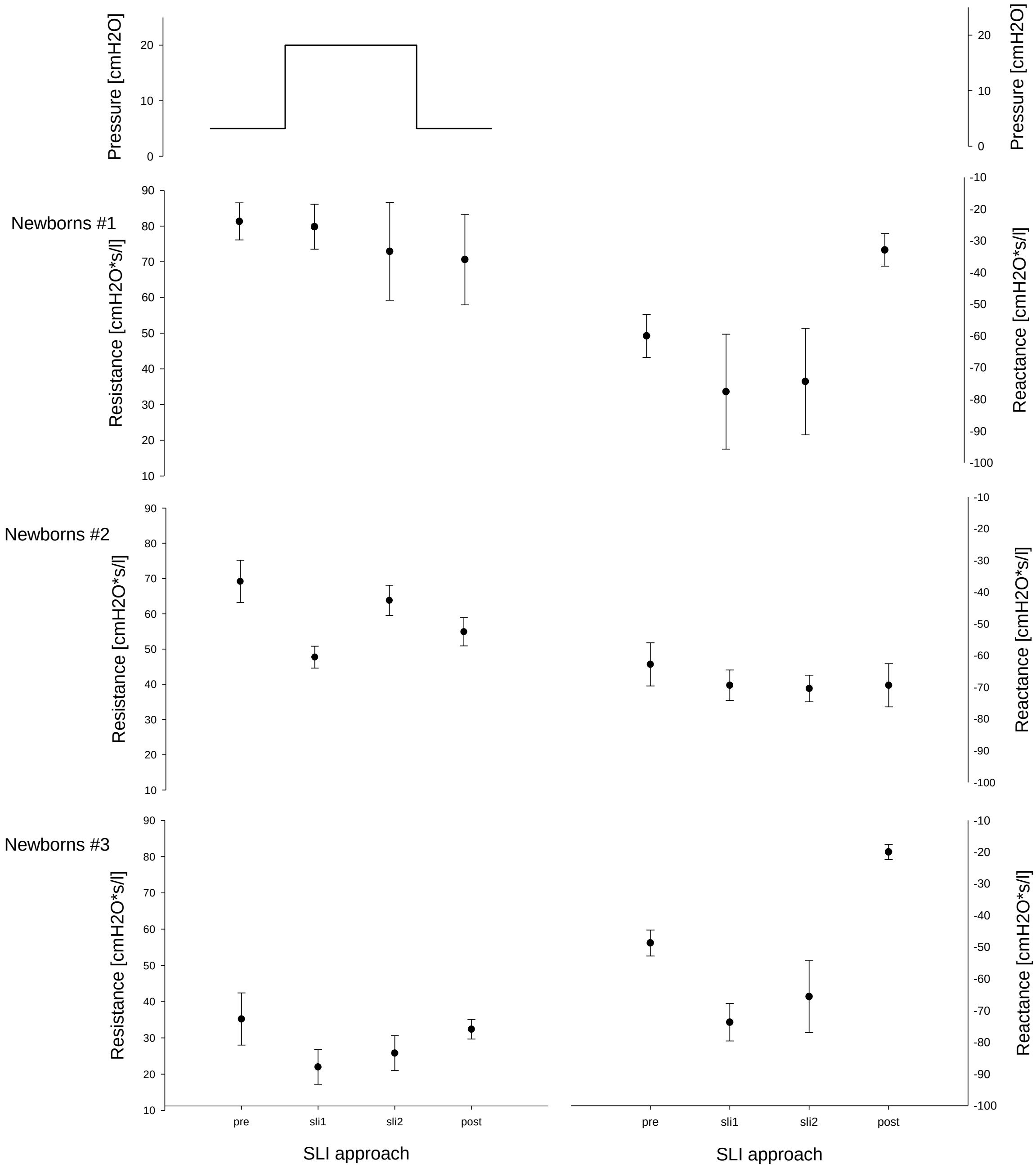
For each recording four parts have been considered:

the *pre SLI* tract, the *during SLI* tract – divided in *first* and *second* part - and *post SLI* tract.

In order to better observe the changes in lung mechanics due to SLI, mean and standard deviation of R and X were computed for each of the previously mentioned tract and for each infant and reported in the following Graph 5. 1. In the upper part of the figure, both on the left and on the right side, a schematic representation of the change in pressure occurring during SLI is reported as reference.

As the lecturer will observe in the next page, Reactance in infants #1 and #3 presents similar behaviors. It decreases during SLI, and increases when the pressure is removed, stabilizing to higher values than before SLI. This behavior is consistent with tissue distension of the already open alveoli and recruitment of collapsed or fluid-filled alveoli during SLI.

This can be explained considering the simplifying hypothesis that the alveolar units are connected to each other in parallel. In this case the measured compliance is the sum of the compliance of the single alveolar units that are reached by the oscillations (see model in the next Figure 5. 3).



Graph 5. 1 Graphs representing resistance and reactance values for each patient. each graph displays four mean and standard deviation values. In the upper part, there is a representation in pressure of what happens during SLI.

When the inflating pressure increases, collapsed alveoli may be opened. The opened alveoli add their compliance to that of already recruited lung area. Thus, total compliance increases, the negative contribution to X_{RS} reduces and consequently X_{RS} increases. In the same time however, the compliance of the already open alveoli decreases, as high PEEP overdistends them and eventually this effect predominates, and the global measured effect is a decrease in X_{RS} value.

When inflation stops, the recruited alveoli may remain open as the closing pressure of an alveolus is lower than its opening pressure, while the compliance of the overdistended alveoli increase again. As a result X_{RS} tends to increase and reaches a higher value than before SLI as a higher number of alveoli are open.

In these infants resistance decreases during SLI, likely due to the increased airways diameter, which is in turn due to the elastic recoil of the lung on the airways walls and recruitment of more alveolar units which, connected in parallel with the already recruited ones, may have reduced the total resistance.

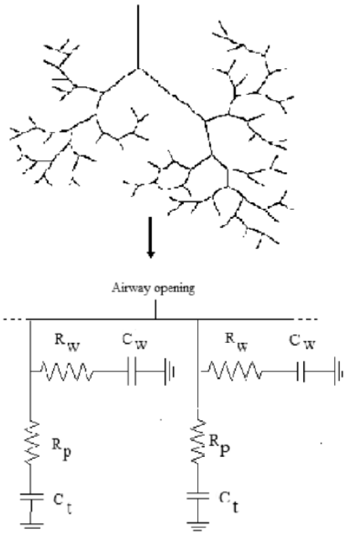


Figure 5. 3 Representation of lung and alveoli electric equivalent model

A different behavior was observed in infant # 2. He does not show signs of recruitment as there is no increase in reactance after SLI. This could mean that the lung is fully recruited since the beginning, but this hypothesis is not supported by the low reactance values, or that the derecruitment is not reversible at least with the pressure and time used in this maneuver.

CONCLUSIONS and FUTURE DEVELOPMENTS

At birth the lung is filled of fetal liquid and the first attempt of breathing has to overcome extremely high inertial and resistive forces to open the lung and to establish an adequate Functional Residual Capacity (FRC).

When the baby at birth is not able to breath and establish an adequate FRC spontaneously, resuscitation manoeuvres aiming at recruiting lung volume, may be necessary.

Currently the most commonly used recruitment strategy at birth is the Sustained Lung Inflation (SLI), which consists in the application of a high positive pressure (20-25 cmH2O) at the airways open, for about 15 seconds. All current approaches to delivering a SLI have focused on applying a clinician pre-determined applied pressure and duration. Such strategies are physiologically incongruous and simplify the complex interaction between the mechanical properties of the diseased lung and the recruitment manoeuvre, exposing the lung to a risk of inadequate or excessive pressure/volume states.

Despite the increasing awareness about the importance of monitoring changes in lung mechanics in newborns during resuscitation, which is the best approach to perform a SLI is still unclear. For sure it would be important to individualize the treatment, because in this critical phase even a mild pulmonary damage, could trigger an inflammatory cascade hardly controllable. Some recent studies have tried to evaluate the effect of SLI matched or not to PEEP. Different values of pressure and time of application have been proposed.

It would be extremely useful to monitor the changes in lung mechanics during these recruiting manoeuvres, in order to help the clinicians to define an optimal approach to increase FRC, avoiding excessive lung tissue stress.

The monitoring in fact may help the clinicians, and so the little patients, in different ways:

- not to provide excessive pressures
- not to provide excessive tidal volume
- observe the level of lung recruitment and of the following de-recruitment

- evaluate a compromise between recruitment and avoidance of excessive stimulation.

In the present study a measuring set-up for monitoring changes in lung mechanics, using FOT and during SLI manoeuvres, has been developed..

In vitro experiments showed that the developed system is able to provide accurate measurements of resistance and reactance on a mechanical analogue of the infant's respiratory system, when compared to an already validated gold standard.

Error between these two measurement in the all the lung tests is lower than 10%, compliant with European FOT guidelines:

Resistance Error % 4.37 ± 3.99 cmH_2O*s/l, Reactance Error % 5.42 ± 4.21 cmH_2O*s/l.

An *in vivo* feasibility study has been performed. Newborns have been measured in nursery room at *Clinica Mangiagalli* in Milan. Newborns were all females. Gestational age is between 30 and 34 weeks: mean gestational age 31.33 ± 2.3 weeks.

Mean age after birth: 18 ± 10.3 days and mean weight at birth: 1195 ± 169 g.

Results observation on the three infants showed that:

- the set-up allows the measurement of respiratory system impedance during SLI in newborns
- the set-up is easy to manage by the clinician during the manoeuvre
- the set-up and procedure is well tolerated by the infants
- the set-up is suitable to be used in the clinical environment.

The criticality that emerged from these first attempts to monitor lung impedance changes during SLI is the interface with the infant particularly in terms of leaks, which should be avoided as much as possible to get reliable measurements.

The preliminary results highlight the difference of lung mechanics response to SLI maneuver between infants. This suggests that monitoring lung mechanics could help in discerning between lungs that benefit or not from SLI and adapt pressure and time of the SLI maneuver according to infants needs.

The measurement system that has been developed with the present work setup represents the set-up necessary to perform further studies aiming at evaluating the clinical benefits of optimizing the treatment according to lung mechanics measured by FOT.

Further Developments

SETUP

A possible future development would be the generation of oscillations by controlling the PEEP valve instead of using the HFOV module.

This would have the advantage of improving the time response of the system and therefore to achieve a steeper increase of the pressure during SLI. In fact, as explained in chapter 4, the use of the HFOV module introduces a big compliance to the system slowing down its mechanical response.

SOFTWARE

Android application could automatically chose stable tracts during the inflation and give a feedback of only reliable values. This automatic algorithm would probably requires not only the flow waveform but also the tidal expiratory volume to best understand the stable tracts. The volume in object could be calculated by the application itself, or delivered by the ventilator through serial communication. In fact the data is already present in Fabian.

PATIENTS

To best test the setup and understand the changes in lung mechanics induced by SLI, a wider number of patients should be observed.

Moreover, this setup can be used at birth, in the delivery room to study how the lung reacts to these manoeuvre in this critical moment.

PROTOCOL

Further studies can address the choice of the best stimulus frequency. Low frequencies are more sensitive to lung peripheral but the Signal to Noise Ratio (SNR) is low because of interference with the frequency components of the breathing signal. In adults 5Hz represents the best compromised between sensitivity to lung periphery and SNR, however as infants have higher breathing frequencies and higher resonant frequency, an higher stimulus frequency could be a best choice.

REFERENCES

1. Horsfild, K., G. Dart, D. E. Olson, and G. Cumming. **Models of the human bronchial tree**. J Appl Physiol. 1971;31:207-17.

2. Kotecha S. **Lung growth for beginners**. Paediatr Respir Rev. 2000;1(4):308-13.

3. Merkus PJ, ten Have-Opbroek AA, Quanjer PH. **Human lung growth: A review**. Pediatr Pulmonol. 1996;21(6):26-36.

4. Guyton AC HJ. **Il sistema respiratorio**. In: Fisiologia Medica. ; 2006.

5. Nagels, J., F. J. Landser, L. Van der Linden, J. Clement, and K.P. Van de Woestijne. **Mechanical properties of lungs and chest wall during spontaneous breathing**. J Appl Physiol. 1980;49:408-16.

6. Miserocchi G DR. *Fisiologia e biofisica medica* - **meccanica ed energetica respiratoria**. 3rd ed. Baldissera; 3rd edition.

7. Harris RS. **Pressure-volume curves of the respiratory system**. Respiratory Care. 2005;50(1).

8. Venegas G HR. **A comprehensive equation for the pulmonary pressure-volume curve**. J Appl Physiol. 1998;84(1):389-95.

9. King RJ. **Pulmonary surfactant**. J Appl Physiol. 1982;53(1):1.

10. Rahn H, Otis AB, Chadwick LE, Fenn WO. **The pressure-volume diagram of the thorax and lung**. Am J Physiol. 1964;146:161-78.

11. Montanini R. **Misure di portata e velocità nei fluidi -** appunti del corso di misure meccaniche e termiche [dissertation].

12. Peslin R FJ. **Oscillation mechanics of the respiratory system**. In: Handbook of Physiology. The Respiratory System. Mechanincs of Breathing. ; 1986. p. 145-77.

13. DuBois AB. **Resistance to breathing**. In: *Handbook of Physiology. Section3,Respiration. Vol 1.* ; 1964. p. 451-62.

14. Oostveen E, MacLeod D, Lorino H, Farre R, Hantos Z, Desager K, Marchal F, ERS. **The forced oscillation technique in clinical practice: methodology, recommendations and future developments**. Eur Respir J. 2003;22(6):1026-41.

15. Dellaca RL, Andersson OM, Zannin E, Kostic P, Pompilio PP, Hedenstierna G et al.: **Lung recruitment assessed by total respiratory system input reactance.** Intensive Care Med 2009.

16. Dellaca RL, Zannin E, Kostic P, Olerud MA, Pompilio PP, Hedenstierna G et al.: **Optimisation of positive end-expiratory pressure by forced oscillation technique in a lavage model of acute lung injury.** Intensive Care Med 2011, 37: 1021-1030.

17. Dellaca RL, Veneroni C, Vendettuoli V, Zannin E, Matassa P, Pedotti A, Colnaghi M, Mosca F In press, **Relationship between respiratory impedance and positive end-expiratory pressure in mechanically ventilated neonates** Intensive Care Med

18. Gauthier R, Beyaert C, Feillet F, Peslin R, Monin P, Marchal F. **Respiratory oscillation mechanics in infants with bronchiolitis during mechanical ventilation**. Pediatr Pulmonol. 1998;25(1):18-31.

19. Stocks J GS. **Specific airway conductance in relation to postconceptional age during infancy**. J Appl Physiol. 1977;49(1):144-54.

20. Cook CD, Cherry RB, O'Brian D, Kalberg P, Smith CA. **Studies of respiratory physiology in the newborn infant**. J Clin Invest. 1955;34:975-82.

21. Allen JL, Wolfson MR, McDowell K, Shaffer TH. **Thoracoabdominal asynchrony in infants with airflow obstruction**. Am Rev Respir Dis. 1990;141(2):337-42.

22. Miller MJ, Carlo WA, DiFiore JM, Martin RJ. **Airway obstruction during periodic breathing in premature infants**. J Appl Physiol. 1988;64(6):2496-500.

23. Gappa M, Pillow JJ, Allen J, Mayer O, Stocks J. **Lung function tests in neonates and infants with chronic lung disease: Lung and chest-wall mechanics**. Pediatr Pulmonol. 2006;41(4):291-317.

24. Garvey J. **Infant respiratory distress syndrome**. Am J of Nursing. 1975;75(4):614.

25. Harling E, Beresford MW, Yoxall CW. **Does sustained lung inflation at resuscitation reduce lung injury in the preterm infant?**. Arch Dis Child. 2005;90(5):406-10.

26. Motoyama EK, Fort MD, Klesh KW, Mutich RL, Guthrie RD. **Early onset of airway reactivity in premature infants with bronchopulmonary dysplasia**. AmRev Respir Dis. 1987;136:50-7.

27. Conti G, Antonelli M, Meduri GU. **Noninvasive vs. conventional mechanical ventilation in patients with chronic obstructive pulmonary disease after failure of medical treatment in the ward: A randomized trial.**. Intensive Care Med. 2002;28(12):1701-7.

28. te Pas AB, siew M, Yagi N et al. **Establishing functional residual capacity at birth: The effect of sustained inflation and positive end-expiratory pressur in rabbit lung model.**. Pediatric Research. 2009;65(5):597-41.

29. te Pas AB, Walther FJ. **A randomized controlled trial of delivery-room respiratory maagement in very preterm infants.** Pediatrics. 2007;120(2):322-9.

30. Kacmarek RM MA. **High-frequency oscillatory ventilation: What large-animal studies have taught us!**. Crit Care Med. 2005;33(3):148-54.

31. Papile L, Burstein L, Burstein R. **Incidence and evolution of subependymal and intraventricular emorrage: A study of infants with birth weight less than 1500gm**. J Pediatrics. 1978;92(4):529-34.

32. Dellaca RL, Rotger M, Aliverti A, Navajas D, Pedotti A, Farre R. **Noninvasive detection of expiratory flow limitation in COPD patients during nasal CPAP**. Eur Respir. 2006;27(5):983.

33. Diagnostic R. **CRIB II: An update of the clinical risk for babies score**. . 2003;361:1789-91.

34. Blanch PB DD. **Chatburn's ventilator classification scheme: A poor substitute for the classic approach**. Respir Care. 1994;39:762.

35. Chatburn RL. **A new system for understanding mechanical ventilators**. Respir Care. 1991;36:1123.

36. Chatburn RL. In: **Classification of mechanical ventilation**. American association for respiratory care; Dallas. ; 1988.

37. Banner MJ. **Expiratory positive pressure valves and work breathing**. Respir Care. 1987;32:431.

38. Mushin WW, Rendell-Baker L, Thompson PW, et al. **Automatic ventilation of the lungs**. FA Davis, editor. Philadelphia: ; 1980.

39. Morris SC FJ. **Transient thermal response of a hot-wire anemometer.** 2003;14:251-9.

40. Li J. **Hot wire attenuation and its correction in turbolence measurements.** 2007:903-6.

41. Anzueto A, Peters JI, Tobin MJ, de los Santos R, Seidenfeld JJ, Moore G, Cox WJ, Coalson JJ. **Effects of prolonged controlled mechanical ventilation on diaphragmatic function in healthy adult baboons**. Crit Care Med. 1997;25(7):1187-90.

42. Baldev R Bhanabahi G. **Optimal PEEP settings in mechanical ventilation usig EIT**. . 2012.

43. Sinha SK DS. **Pressure support ventilation**. In: Manual of Neonatal Respiratory Care. ; 2000. p. 157-60.

44. Williams P MM. **Pressure support and pressure assist/control: Are there differences?: An evaluation of the newest intensive care unit ventilators**. Respir Care. 2000;45:11.

45. Connors AF, McCaffree DR, Gray BA. **Effect of inspiratory flow rate on gas exchange during mechanical ventilation**. Am Rev Respir Dis. 1981;124(5):537-43.

46. Navajas D FR. **Forced oscillation assessment of respiratory mechanics in ventilated patients**. Crit Care. 2001;5(1):3-9.

47. van Genderingen HR, Heethaar RM. **Reduction of oscillatory pressure along the endotracheal tube is indicative for maximal respiratory compliance during high-frequency oscillatory ventilation: A mathematical model study**. Pediatr Pulmonol. 2001;31(6):458-63.

48. Weber K, Courtney SE, Pyon KH, Chang GY, Pandit PB, Habib RH. **Detecting lung overdistention in newborns treated with high-frequency oscillatory ventilation**. J Appl Physiol. 2000;89(1):364-72.

49. Henderson-Smart DJ, Cools F, Bhuta T, Offringa M. **Elective high frequency oscillatory ventilation versus conventional ventilation for acute pulmonary dysfunction in preterm infants**. Cochrane Database Syst Rev. 2007;3(3).

50. Burd B. **Beginning programing in java for dummies**. 2nd ed. Indianapolis: Wiley Publishing, Inc.

51. Burd B. **Android application development for dummies**.

APPENDIX

APPENDIX A Acronyms

Physical measures

VT	Tidal Volume	[l]
VD	Dead Volume	[l]
IRV	Inspiratory Reserve Volume	[l]
ERV	Expiratory Reserve Volume	[l]
RV	Residual Volume	[l]
TLC	Total Lung Capacity	[l]
VC	Vital Capacity	[l]
FRC	Functional Residual Capacity	[l]
IC	Inspiratory Capacity	[l]
PBS	Body Surface Pressure	[cmH2O]
PAO	Airways Opening Pressure	[cmH2O]
PALV	Alveolar Pressure	[cmH2O]
PPL	Pleural Pressure	[cmH2O]
PTR	Transpulmonary Pressure	[cmH2O]
PW	Wall Pressure	[cmH2O]
PRS	Respiratory System Pressure	[cmH2O]
PAW	Airways Pressure Drop	[cmH2O]
PRES	Pressure due to Resistance	[cmH2O]
PEL,L	Elastic Pressure of the Lung	[cmH2O]
PEL,CW	Elastic Pressure of the Chest Wall	[cmH2O]
PMUS	Pressure generated by the respiratory Muscles	[cmH2O]
PES	Esophageal Pressure	[cmH2O]
$\dot{V}$	Flow	[l/s]
γ	Surface tension	[J/ m^2]
ID	Internal Diameter	[m]
CRS	Compliance of the total Respiratory System	[l/cmH2O]
CL	Lung Compliance	[l/cmH2O]
CCW	Chest Wall Compliance	[l/cmH2O]
C_T	Tissue Compliance	[l/cmH2O]
E_{RS}	Elastance of the Respiratory System	[cmH2O/l]
R_{RS}	Resistance of the Respiratory System	[cmH2O*s/l]
RAW	Airway Resistance	[cmH2O*s/l]
RT	Tissue Resistance	[cmH2O*s/l]
IRS	Inertance of the Respiratory System	[cmH2O*s^2/l]
ZIN or ZRS,IN	Input impedance of the Respiratory System	[cmH2O*s/l]
ZTR or ZRS,TR	Transfer Impedance	[cmH2O*s/l]
ZAW	Airways Impedance	[cmH2O*s/l]
XRS	Reactance	[cmH2O*s/l]
ω	Pulsation	[rad/s]

Clinical and technical acronyms

BPD	BronchoPulmonary Dysplasia	[]
GA	Gestational Age	[]
PEEP	Positive End Expiratory Pressure	[cmH2O]
PIP	Peak Inspiratory Pressure	[cmH2O]
I:E	Inspiratory-expiratory time ratio	[]
TI	Inspiratory Time	[s]
TE	Expiratory Time	[s]
IPPV	Intermittent Positive Pressure Ventilation	[]
IMV	Intermittent Mandatory Ventilation	[]
SIPPV	Synchronized Intermittent Positive Press Ventilation	[]
SIMV	Synchronized Intermittent Mandatory Ventilation	[]
PSV	Pressure Support Ventilation	[]
PSV-SIPPV	Pressure Support Ventilation- Synchronized Intermittent Positive Pressure Ventilation	[]
PSV-SIMV	Pressure Support Ventilation- Synchronized Intermittent Mandatory Ventilation	[]
CPAP	Continuous Positive Airway Pressure	[]
BiPAP	Bilevel Positive Airway Pressure	[]
IPAP	Inspiratory Pressure	[cmH2O]
EPAP	Expiratory Pressure	[cmH2O]
NIV	Non Invasive Ventilation	[]
HFOV	High Frequency Oscillation Ventilation	[]
VILI	Ventilatory Induced Lung Injury	[]
FFT	Fast Fourier Transform	[]
LMSM	Least Mean Square Method	[]
S	Vector - Measured samples	[]
X	Vector - Fourier coefficients	[]
PSD	Power Spectral Density	[]

Communication and programming

DTE	Data Terminal Equipment	[]
DCE	Data Communication Equipment	[]
USB	Universal Serial Bus	[]
RTS	Request To Send	[]
CTS	Clear To Send	[]
DTR	Data Terminal Ready	[]
DSR	Data Set Ready	[]
OTG	On The Go - USB converter cable	[]
OOP	Object Oriented Programming	[]
JVM	Java Virtual Machine	[]
IDE	Integrated Development Environment	[]
JTD	Java Development Tool	[]
SDK	Software Development Kit	[]
API	Android Programming Interface	[]

APPENDIX B Android Code

The hierarchy of the Android application has been explained, such as the rationale. In this section the main files are simply listed.

B.1 Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
android:orientation="vertical"
     >

                <TableLayout
                    android:layout_width="1200dp"
                    android:layout_height="85dp" >

                    <TableRow
                        android:id="@+id/tableRow1"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content" >
                                            <TextView
                               android:id="@+id/status"
                               android:layout_width="wrap_content"
                               android:layout_height="wrap_content"
                               android:height="200dp"
                               android:text="@string/hello"
                               android:width="500dp" />


                           <Button
                               android:id="@+id/button"
                               android:layout_width="wrap_content"
                               android:layout_height="wrap_content"
                               android:layout_gravity="right"
                               android:height="80dp"
                               android:onClick="recordData"
                               android:text="@string/button_go"
                               android:visibility="visible"
                               android:width="200dp" />
                    </TableRow>

                </TableLayout>


     <ScrollView
            android:id="@+id/scrollView1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >

            <LinearLayout
```

```
                android:layout_width="match_parent"
                android:layout_height="match_parent"
   android:orientation="vertical">


        <com.androidplot.xy.XYPlot
            android:id="@+id/plot1"
            android:layout_width="1200dp"
            android:layout_height="290dp"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            android:layout_marginTop="10dp"
            title="PRESS Plot" />

        <com.androidplot.xy.XYPlot
            android:id="@+id/plot2"
            android:layout_width="1200dp"
            android:layout_height="290dp"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            android:layout_marginTop="10dp"
            title="FLOW Plot" />

        <com.androidplot.xy.XYPlot
            android:id="@+id/plot3"
            android:layout_width="1200dp"
            android:layout_height="290dp"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            android:layout_marginTop="10dp"
            title="R Plot" />

        <com.androidplot.xy.XYPlot
            android:id="@+id/plot4"
            android:layout_width="1200dp"
            android:layout_height="290dp"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            android:layout_marginTop="10dp"
            title="X Plot" />
            </LinearLayout>
        </ScrollView>

</LinearLayout>
```

B.2 Manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.hoho.android.usbserial.examples"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="12" />
    <uses-feature android:name="android.hardware.usb.host" />
       <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <application
            android:hardwareAccelerated="false"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name="com.hoho.android.usbserial.examples.DemoActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.DEFAULT" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <intent-filter>
                <action

android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
            </intent-filter>
            <meta-data
                android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
                android:resource="@xml/device_filter" />
        </activity>
    </application>

</manifest>
```

B.3 DemoActivity.java

```
package com.hoho.android.usbserial.examples;
import android.app.Activity;
import android.content.Context;
import android.graphics.Color;
import android.hardware.usb.UsbManager;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import com.androidplot.series.XYSeries;
import com.androidplot.xy.*;
import com.hoho.android.usbserial.R;
import com.hoho.android.usbserial.driver.UsbSerialDriver;
import com.hoho.android.usbserial.driver.UsbSerialProber;
import com.hoho.android.usbserial.examples.Model.DataPacket;
import com.hoho.android.usbserial.util.HexDump;
import com.hoho.android.usbserial.util.SerialInputOutputManager;
import com.hoho.android.usbserial.examples.Model;
import com.hoho.android.usbserial.examples.SaveData;

import java.io.IOException;
import java.util.Arrays;
import java.util.LinkedList;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

    private UsbManager mUsbManager;
    private TextView status;
    private TextView val1Text;
    private TextView val2Text;
    private Button button;

    private          final          ExecutorService          mExecutor          =
Executors.newSingleThreadExecutor();
    private SerialInputOutputManager mSerialIoManager;
    private final SerialInputOutputManager.Listener mListener =
            new SerialInputOutputManager.Listener() {
        @Override
        public void onRunError(Exception e) {
            Log.d(TAG, "Runner stopped.");
        }
        @Override
        public void onNewData(final byte[] data) {
            DemoActivity.this.runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    DemoActivity.this.updateReceivedData(data);
                }
            });
        }
    };
    private SaveData save;
```

```
    private Model model;
    private CPU cpu;
    private int domBound = 100;   //non >100!!
    private int demultdivisor = 5;   //non >10!!   best: 80,8   - 90,10  -
100,10!!
    private XYPlot plot1;
    private XYPlot plot2;
    private XYPlot plot3;
    private XYPlot plot4;
    private LinkedList pointsHistory1 = new LinkedList<Number>();
    private LinkedList pointsHistory2 = new LinkedList<Number>();
    private LinkedList pointsHistory3 = new LinkedList<Number>();
    private LinkedList pointsHistory4 = new LinkedList<Number>();
    private SimpleXYSeries Points1 =  new SimpleXYSeries("PRESSURE");
    private SimpleXYSeries Points2=  new SimpleXYSeries("FLOW");
    private SimpleXYSeries Points3 =  new SimpleXYSeries("RESISTANCE");
    private SimpleXYSeries Points4 =  new SimpleXYSeries("REACTANCE");

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        status = (TextView) findViewById(R.id.status);
        button = (Button) findViewById(R.id.button);
//     Dichiaro i 4 grafici
        plot1 = (XYPlot) findViewById(R.id.plot1);
        plot2 = (XYPlot) findViewById(R.id.plot2);
        plot3 = (XYPlot) findViewById(R.id.plot3);
        plot4 = (XYPlot) findViewById(R.id.plot4);

        mUsbManager = (UsbManager) getSystemService(Context.USB_SERVICE);
        save = new SaveData(this);
        model= new Model(this);
//     Setto i 4 grafici

//PLOT1
        plot1.setLayerType(View.LAYER_TYPE_SOFTWARE, null);
     // setup the APR History plot:
        plot1.setRangeBoundaries(0, 30, BoundaryMode.FIXED);
        plot1.setDomainBoundaries(0, domBound, BoundaryMode.FIXED);
        plot1.addSeries(Points1,         LineAndPointRenderer.class,         new
LineAndPointFormatter(Color.rgb(100, 100, 200), Color.BLACK,null));
        plot1.setDomainStepValue(11);
        plot1.setTicksPerRangeLabel(1);
        plot1.setDomainLabel("Sample Index");
        plot1.getDomainLabelWidget().pack();
        plot1.setRangeLabel("Press (cmH2O)");
        plot1.getRangeLabelWidget().pack();
        plot1.disableAllMarkup();
  //PLOT2
        plot2.setLayerType(View.LAYER_TYPE_SOFTWARE, null);
     // setup the APR History plot:
        plot2.setRangeBoundaries(-6, 6, BoundaryMode.FIXED);
        plot2.setDomainBoundaries(0, domBound, BoundaryMode.FIXED);
        plot2.addSeries(Points2,         LineAndPointRenderer.class,         new
LineAndPointFormatter(Color.rgb(100, 100, 200), Color.BLACK,null));
          plot2.setDomainStepValue(11);
```

```
        plot2.setTicksPerRangeLabel(1);
        plot2.setDomainLabel("Sample Index");
        plot2.getDomainLabelWidget().pack();
        plot2.setRangeLabel("Flow (L/min)");   //check!!
        plot2.getRangeLabelWidget().pack();
        plot2.disableAllMarkup();

//PLOT3
        plot3.setLayerType(View.LAYER_TYPE_SOFTWARE, null);
        plot3.setRangeBoundaries(0, 200, BoundaryMode.FIXED);
        plot3.setDomainBoundaries(0, domBound, BoundaryMode.FIXED);
        plot3.addSeries(Points3,         LineAndPointRenderer.class,         new
LineAndPointFormatter(Color.rgb(100, 100, 200), Color.BLACK,null));
        plot3.setDomainStepValue(11);
        plot3.setTicksPerRangeLabel(1);
        plot3.setDomainLabel("Sample Index");
        plot3.getDomainLabelWidget().pack();
        plot3.setRangeLabel("R");
        plot3.getRangeLabelWidget().pack();
        plot3.disableAllMarkup();

//PLOT4
        plot4.setLayerType(View.LAYER_TYPE_SOFTWARE, null);
        plot4.setRangeBoundaries(-200, 0, BoundaryMode.FIXED);
        plot4.setDomainBoundaries(0, domBound, BoundaryMode.FIXED);
        plot4.addSeries(Points4,         LineAndPointRenderer.class,         new
LineAndPointFormatter(Color.rgb(100, 100, 200), Color.BLACK,null));
        plot4.setDomainStepValue(11);
        plot4.setTicksPerRangeLabel(1);
        plot4.setDomainLabel("Sample Index");
        plot4.getDomainLabelWidget().pack();
        plot4.setRangeLabel("X");
        plot4.getRangeLabelWidget().pack();
        plot4.disableAllMarkup();
    }

    @Override
    protected void onPause() {
        super.onPause();
        stopIoManager();
        if (mSerialDevice != null) {
            try {
                mSerialDevice.close();
            } catch (IOException e) {
                // Ignore.
            }
            mSerialDevice = null;
        }
    }

    @Override
    protected void onResume() {
        super.onResume();
        mSerialDevice = UsbSerialProber.acquire(mUsbManager);
        if (mSerialDevice == null) {
            status.setText("Nessun device connesso.");
        } else {
```

```
            try {
                mSerialDevice.open();
            } catch (IOException e) {
                status.setText("Errore connessione device: " + e.getMessage());
                try {
                    mSerialDevice.close();
                } catch (IOException e2) {
                    // Ignore.
                }
                mSerialDevice = null;
                return;
            }
            status.setText("Device connesso.");
            //ogni volta che attacco un device diverso, azzero il numero di
file
            save.resetFileNumber();
        }
        onDeviceStateChange();
    }
    private void stopIoManager() {
        if (mSerialIoManager != null) {
            mSerialIoManager.stop();
            mSerialIoManager = null;
        }
    }

    private boolean isRecording=false;
    public void recordData(View _view){
        if(!isRecording){
            save.resetFileNumber();
            try {
                save.createWriter();
            } catch (Exception e) {
                //status.setText("Errore, impossibile creare il file");
                return;
            }
           //ogni volta che avvio il listener, azzero il numero di file
            mExecutor.submit(mSerialIoManager);
            isRecording=true;
            button.setText("STOP");
            cpu = new CPU(save);  //si alloca ad ogni "REC" giusto??
        }else{
            mSerialIoManager.stop();
            mSerialIoManager   =   new   SerialInputOutputManager(mSerialDevice,
mListener);
            isRecording=false;
            button.setText("REC");
            error("NO RECORDING");
            save.closeWriter();
        }
    }
    private void startIoManager() {
        if (mSerialDevice != null) {
            mSerialIoManager   =   new   SerialInputOutputManager(mSerialDevice,
mListener);
        }
    }
```

```
    private void onDeviceStateChange() {
        stopIoManager();
        startIoManager();
    }
    private void updateReceivedData(byte[] data) {
        try{
        model.addString(HexDump.dumpString(data));
        }catch(Exception ex){
            return;
        }
    }
private int demultiplexer=1;
    public void write(DataPacket p) {
        save.write(cpu.calcola(p));
        if(demultiplexer%demultdivisor==0){
        if (pointsHistory1.size() > domBound) {     //controllo solo uno tanto
il packet è restituito di tt e 4 (eccetto a fine REC)
            pointsHistory1.removeFirst();
            pointsHistory2.removeFirst();
            pointsHistory3.removeFirst();
            pointsHistory4.removeFirst();
        }
        // add the latest history sample:
        pointsHistory1.addLast(p.val1);
        pointsHistory2.addLast(p.val2*60);
        pointsHistory3.addLast(cpu.Rrs);
        pointsHistory4.addLast(cpu.Xrs);
        // update the plot with the updated history Lists:
        Points1.setModel(pointsHistory1,
SimpleXYSeries.ArrayFormat.Y_VALS_ONLY);
        Points2.setModel(pointsHistory2,
SimpleXYSeries.ArrayFormat.Y_VALS_ONLY);
        Points3.setModel(pointsHistory3,
SimpleXYSeries.ArrayFormat.Y_VALS_ONLY);
        Points4.setModel(pointsHistory4,
SimpleXYSeries.ArrayFormat.Y_VALS_ONLY);
        // redraw the Plots:
        plot1.redraw();
        plot2.redraw();
        plot3.redraw();
        plot4.redraw();
        demultiplexer=1;
        }else{
         demultiplexer++;
        }
    }
    public void error(String s){
        status.setText(s);
    }
}
```

B.4 Model.java

```
package com.hoho.android.usbserial.examples;
public class Model {
    private StringBuilder buffer = new StringBuilder();
    private DemoActivity save;
    public Model(DemoActivity s){
        save=s;
    }

  //public int numOfPack=0;
    public synchronized void addString(String data){
        //se si tenta di aggiungere altre stringhe mentre questo
        //sta lavorando, si mette in coda
        synchronized(buffer){
            buffer.append(data);
            analize();
        }
    }
    private void analize(){
        String temp = buffer.toString();
        boolean read=false;
        //va avanti a cercare marker
        while(temp.indexOf("#")!=-1){
            //se entra nel ciclo vuol dire che ne ha trovato uno
            //seziono la stringa per eliminare il primo
            String temp2=temp.substring(temp.indexOf("#")+1);
            //cerco un secondo marker, così so dove finisce il pacchetto di
dati
            if(temp2.indexOf("#")!=-1){
                //se trovo il secondo marker
                //estraggo i dati
                //e vado avanti a lavorare sulla stringa senza i dati appena
letti
                String data = temp2.substring(0,temp2.indexOf("#"));
                readPacket(data);
                temp=temp2.substring(temp2.indexOf("#"));
                read=true;
            }else{
                //se non trovo il secondo marker vuol dire che devo aspettare
altri dati
                //quindi mi fermo e aspetto
                break;
            }
        }
        //Se ho letto qualcosa dal buffer, aggiorno il buffer
        if(read){
            buffer = new StringBuilder();
            buffer.append(temp);
        }
    }
    //0|0|0|0|0|0|0|0|0|0|0|0|0|0|79|0|59|0|31|0
    private void readPacket(String data){
        String[] temp=data.split("\\|");
        for(int i=0;i<temp.length-1;i=i+2){
            try{
```

```
                DataPacket p =new DataPacket(temp[i],temp[i+1]);
                save.write(p);
            }catch(Exception e){}
        }
    }
public class DataPacket{
    public double val1;
    public double val2;
    public DataPacket(String v1, String v2){
        if((v1.indexOf("A`")!=-1 || v2.indexOf("A`")!=-1)){
//segnalo l'errore    oppure:
            v1=v1.replace("A`","");
            v2=v2.replace("A`","");
            //non faccio nulla!!
        }
            val1=Integer.valueOf(v1) * 0.1;
            val2=Integer.valueOf(v2)* 0.000016667;
    }
    @Override
    public String toString(){
        return String.valueOf(String.format("%.4f",
val1))+"\t"+String.valueOf(String.format("%.8f", val2));
    }
}
}
```

B.5 CPU.java

```
package com.hoho.android.usbserial.examples;
import java.util.ArrayList;
import com.hoho.android.usbserial.examples.Model.DataPacket;
public class CPU {
    private ArrayList<DataPacket> dataPacketList = new ArrayList<DataPacket>();
    private SaveData save;
    //    private int k=0;
    private int window=40;
    private double Ap= 0; //cos' riinizializzo a "0" e poi lancio i calcoli!
    private double Af= 0;
    private double Bp= 0;
    private double Bf= 0;
    private double Z_den;
    private double Z_num;
    private double Z;
    private double Z_abs;
    private double Z_phi;
    private double Z_phnum;
    private double Z_phden;
    public double Rrs;
    public double Xrs;
    private DataPacket actualPacket;
    private double[][] mB = matriceB();
    private String valoriRX;
    public CPU(SaveData s) {
        save=s;
    }
//public void calcola(DataPacket p) {
     public String calcola(DataPacket p) {
        dataPacketList.add(p);
        if(dataPacketList.size()==41){
            dataPacketList.remove(0);
            actualPacket = dataPacketList.get(20);
            valoriRX= conti();
        }else if( dataPacketList.size()==40){
            valoriRX = conti();
            actualPacket = dataPacketList.get(20);
       }else{
           valoriRX="errore";
       }
return (actualPacket.toString()+"\t"+String.valueOf(valoriRX)+"\t");
    }
    private String conti(){
        double Ap= 0;
        double Bp=0;
        double Af=0;
        double Bf=0;
double P=0;
double F=0;
         for(int i=0;i<(window);i++) { //controlla indice di matrici se inizia
ad 1 o 0!!
           DataPacket actual =dataPacketList.get(i);
            Ap = Ap + mB[0][i]* actual.val1;
            Bp = Bp + mB[1][i]* actual.val1;
```

```
            Af = Af + mB[0][i]* actual.val2;
            Bf = Bf + mB[1][i]* actual.val2;
        }
        Rrs=5;
        Xrs=Bp+3.5;

        Z_den= Af*Af + Bf*Bf;
        Z_num= Ap*Ap + Bp*Bp;
        Z_abs=Math.sqrt(Z_num/Z_den);
        Z_phden= Math.atan2(Bf,Af);
        Z_phnum= Math.atan2(Bp,Ap);
        Z_phi = Z_phnum - Z_phden;
if((Z_phi)> Math.PI){
    Z_phi =Z_phi -2*Math.PI;
}else if(Z_phi < -Math.PI){
    Z_phi =2*Math.PI - Z_phi;
}
 //       Rrs=Z_abs*1.0649*Math.cos(Z_phi-(-0.0187)*5);  //correction "CPAP"
   //     Xrs=Z_abs*1.0649*Math.sin(Z_phi-(-0.0187)*5);  //correction "CPAP"
Rrs=Z_abs*Math.cos(Z_phi-(-0.1881)*5); //correction "HFO"
Xrs=Z_abs*Math.sin(Z_phi-(-0.1881)*5);  //correction "HFO"
 return
String.valueOf(String.format("%.5f",Rrs))+"\t"+String.valueOf(String.format("%.
5f", Xrs));
}
private double[][] matriceB(){

    double[][] matrixBnostra= new double[2][40];
    matrixBnostra[0][0] = 0.05;
    matrixBnostra[0][1] = 0.0493844170297569;
    matrixBnostra[0][2] = 0.0475528258147577;
    matrixBnostra[0][3] = 0.0445503262094184;
    matrixBnostra[0][4] = 0.0404508497187474;
    matrixBnostra[0][5] = 0.0353553390593274;
    matrixBnostra[0][6] = 0.0293892626146237;
    matrixBnostra[0][7] = 0.0226995249869773;
    matrixBnostra[0][8] = 0.0154508497187474;
    matrixBnostra[0][9] = 0.00782172325201155;
    matrixBnostra[0][10] = 5.14328516904055E-18;    //
    matrixBnostra[0][11] = -0.00782172325201154;
    matrixBnostra[0][12] = -0.0154508497187474;
    matrixBnostra[0][13] = -0.0226995249869773;
    matrixBnostra[0][14] = -0.0293892626146237;
    matrixBnostra[0][15] = -0.0353553390593274;
    matrixBnostra[0][16] = -0.0404508497187474;
    matrixBnostra[0][17] = -0.0445503262094184;
    matrixBnostra[0][18] =  -0.0475528258147577;
    matrixBnostra[0][19] = -0.0493844170297569;
    matrixBnostra[0][20] = -0.05;
    matrixBnostra[0][21] = -0.0493844170297569;
    matrixBnostra[0][22] = -0.0475528258147577;
    matrixBnostra[0][23] = -0.04455032620941840;
    matrixBnostra[0][24] = -0.0404508497187474;
    matrixBnostra[0][25] = -0.0353553390593274;
    matrixBnostra[0][26] = -0.0293892626146237;
    matrixBnostra[0][27] = -0.0226995249869773;
    matrixBnostra[0][28] = -0.0154508497187474;
```

```
    matrixBnostra[0][29] = -0.00782172325201155;
    matrixBnostra[0][30] = 3.17546230394475E-17;   //
    matrixBnostra[0][31] = 0.00782172325201154;
    matrixBnostra[0][32] = 0.0154508497187474;
    matrixBnostra[0][33] = 0.0226995249869773;
    matrixBnostra[0][34] = 0.0293892626146237;
    matrixBnostra[0][35] = 0.0353553390593273;
    matrixBnostra[0][36] = 0.0404508497187474;
    matrixBnostra[0][37] = 0.0445503262094184;
    matrixBnostra[0][38] = 0.0475528258147577;
    matrixBnostra[0][39] = 0.0493844170297569;
    matrixBnostra[1][0] = -4.15333634234434E-18;   //
    matrixBnostra[1][1] = -0.00782172325201154;
    matrixBnostra[1][2] = -0.0154508497187474;
    matrixBnostra[1][3] = -0.0226995249869773;
    matrixBnostra[1][4] = -0.0293892626146236;
    matrixBnostra[1][5] = -0.0353553390593274;
    matrixBnostra[1][6] = -0.0404508497187474;
    matrixBnostra[1][7] = -0.0445503262094184;
    matrixBnostra[1][8] =-0.0475528258147577;
    matrixBnostra[1][9] = -0.0493844170297569;
    matrixBnostra[1][10 ]= -0.05;
    matrixBnostra[1][11] = -0.0493844170297569;
    matrixBnostra[1][12] = -0.0475528258147577;
    matrixBnostra[1][13] = -0.0445503262094184;
    matrixBnostra[1][14] = -0.0404508497187474;
    matrixBnostra[1][15] = -0.0353553390593274;
    matrixBnostra[1][16] = -0.0293892626146237;
    matrixBnostra[1][17] = -0.0226995249869773;
    matrixBnostra[1][18] = -0.0154508497187474;
    matrixBnostra[1][19] = -0.0078217232520115;
    matrixBnostra[1][20] = -4.73545521495532E-18;   //
    matrixBnostra[1][21] = 0.00782172325201154;
    matrixBnostra[1][22] = 0.0154508497187474;
    matrixBnostra[1][23] = 0.0226995249869774;
    matrixBnostra[1][24] = 0.0293892626146236;
    matrixBnostra[1][25] = 0.0353553390593273;
    matrixBnostra[1][26] = 0.0404508497187474;
    matrixBnostra[1][27] = 0.0445503262094184;
    matrixBnostra[1][28] = 0.0475528258147577;
    matrixBnostra[1][29] = 0.0493844170297569;
    matrixBnostra[1][30] = 0.05;
    matrixBnostra[1][31] = 0.0493844170297569;
    matrixBnostra[1][32] = 0.0475528258147577;
    matrixBnostra[1][33] = 0.0445503262094184;
    matrixBnostra[1][34] = 0.0404508497187474;
    matrixBnostra[1][35] = 0.0353553390593274;
    matrixBnostra[1][36] = 0.0293892626146237;
    matrixBnostra[1][37] = 0.0226995249869773;
    matrixBnostra[1][38] = 0.0154508497187474;
    matrixBnostra[1][39] = 0.00782172325201156;

    return matrixBnostra;
}
}
```

B.6 SaveData.java

```
package com.hoho.android.usbserial.examples;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Timer;
import java.util.TimerTask;
import android.os.Environment;
public class SaveData {
    private FileWriter outwriter = null;
    private Integer fileNumber = 0;
    private Timer timer1;
    private TimerAction task;
    private DemoActivity demo;
    private String index = "";
    public SaveData(DemoActivity d) {
        demo = d;
    }
    public void createWriter() throws Exception {
        timer1 = new Timer();
        task = new TimerAction();
        newFile();
    }

    private class TimerAction extends TimerTask {
        @Override
        public void run() {
            // TODO Auto-generated method stub
            try {
                newFile();
            } catch (Exception e) {
            }
        }
    }
    public void closeWriter() {
        try {
            outwriter.flush();
            outwriter.close();
            timer1.cancel();
        } catch (Exception e) {
            demo.error("errore timer");
            return;
        }
    }
    public void newFile() throws Exception {
        if (outwriter == null) {
        } else {
            synchronized (outwriter) {
            }
        }
        createNewFile();
```

```
        fileNumber++;
    }
    private void createNewFile() throws Exception {
        Date date = new Date();
        DateFormat dateFormatFile = new SimpleDateFormat("yyyyMMddHHmm");
        DateFormat dateFormatFolder = new SimpleDateFormat("yyyyMMdd");
        String  path  =  Environment.getExternalStorageDirectory().getPath()  +
"/SLI-Thesis_IncomingData/"
                + dateFormatFolder.format(date);
        File pfile = new File(path);
        if (!pfile.exists()) {
            pfile.mkdirs();
        }
        if (fileNumber == 0) {
            index = dateFormatFile.format(date);
          timer1.scheduleAtFixedRate(task, 0, 420000);   //=420K = 7min
      }
    String fname = index + "-part" + fileNumber.toString() + ".txt";
        demo.error("RECORDING    Scrittura dati su file: " + index);
        File outFile = new File(path, fname);
        outwriter = new FileWriter(outFile);
    }
    public void write(String data) {
        try {
            outwriter.write(data + "\n");
            outwriter.flush();
        } catch (Exception e) {
            return;
        }
    }
    public void resetFileNumber() {
        fileNumber = 0;
    }
}
```

RINGRAZIAMENTI

Dal Momento che le cose più care e sentite si esprimono meglio nella propria lingua madre….

Un grazie di cuore a tutti coloro che in questi anni mi sono stati vicini, mi hanno supportato, mi hanno spronato, mi hanno aiutato e mi hanno sfidato!

Grazie a Mamma e Papà, perché mi avete permesso di arrivare fin qui con tutto il sostegno che potevate darmi. Perché con i vostri aiuti e consigli mi avete reso quello che sono e che sarò, e ciò va ben oltre questo traguardo che comunque sento come stupendo e che mi riempie di orgoglio (altro che la triennale!) Grazie!

Grazie Marco ed Alice, i mie due fratellini, seppur ormai cresciuti, a cui ho dedicato la tesi. Si scherza sempre e si cresce insieme e spero di esservi d'esempio sempre, in tutto e per tutto. E poooi,pooooi.. si fabbordellooo!

Grazie a te Veronica, non solo amata, ma anche amica ed esempio. Perché dopo ormai 8 anni si condividono un sacco di cose ognuno è quel che è, grazie all'altro. Sono entusiasta di vivere insieme i primi passi nel modo del lavoro, che ci daranno sicuramente grandi soddisfazioni. E se lavoro già da sei mesi, è perché le tue ansie mi hanno spinto a cercarlo! Ti auguro questa nuova avventura che hai appena intrapreso sia meravigliosa e ti dia quel che meriti. Se sono quello che sono è anche grazie a te!

Un grazie a tutta la mia famiglia:

Grazie alla Nonna, ed agli altri Nonni ormai lontani, che da sempre vi siete presi cura di noi e che spero siate orgogliosi.

Grazie agli zii: Emilia, Loredana ed Antonio che vi premurate sempre e mi consigliate da quando sono piccolo.

Grazie ad Anna, la cugina con cui ho condiviso un sacco di anni ed esperienze, passando dai litigi infantili alle confidenze adulte.

Un grazie ai più nuovi Emmina e Fabio, ormai fate parte della famiglia e sono felice di condividere questo stupendo momento con voi.

Grazie ad Alberto, Maria Grazia e Giulia per il benvenuto e la grande considerazione di me.

Grazie a tutti i parenti cari, zia Miglia, Gianna e Silvio…

Grazie agli amici di sempre:

Grazie a Moreno, sempre nel cuore e sempre amico nonostante spesso ci si perda per tanto tempo.

Grazie a "quelli di Arma":

grazie Zimò, fai sempre sembrare tutto scherzoso e non importante ma so quanto impegno metti in tutto ciò che fai, e non vedo l'ora di festeggiare tutti i tuoi traguardi. Con un Sambuchino?

Grazie Benny, con i consigli ed il supporto che ci diamo. Ti ammiro molto per come sei e per quanto sei forte. Però davvero, ORA basta con la Redbull!

Grazie Stefa e grazie Didi, forse tra tutti "noi" quelli che conosco da meno tempo (beh, se 8 anni circa saranno mai pochi…) ma siete due persone fantastiche e so che su di voi si potrà sempre contare!

Grazie Mat e grazie Eli. Veramente voi agli antipodi: la prima conosciuta e l'ultimo! Ma vi voglio un gran bene allo stesso modo e sono felice di aver passato insieme così tanti bei momenti.

Siamo un grande gruppo e spero la vita non ci separi mai.

Grazie ai cari amici conosciuti in questi anni. Ultimi nel tempo e mai nell'affetto:

Grazie Serena, perché la piccola rivalità che, diciamo la verità, c'è tra noi mi ha sempre spronato. Ma sei un'amica e spero si continuerà a vederci.. abitiamo a soli 20 minuti…

Grazie Marta; mi piace la nostra amicizia ed il tuo carattere. Credo molto in te e devi farlo anche tu!! Gli ultimi traguardi ti esortano a farlo!

Grazie Veronica, sempre di buon umore e cara. Una delle prime compagne trovate e mai allontanata. Spero si continuerà così.

Grazie Roby, ti ammiro molto per i tuoi impegni ed il tuo modo di essere! Non cambiare mai!

Dopo le donne…

Grazie Veeecchiii, grande amico! Conosciuto un po' dopo gli altri (troppo casinista prima), ma subito entrato nel cuore. Voto 8.5 – Mostro di Firenze.

Grazie puer al babbeo Mezza, con la felpa verde. Compagno di scuola, amico di tesi (triennale) e, quasi, di paese! Con te sicuramente non ci si perderà mai di vista! (PS visto che l'ho messo!?)

Grazie al polpo Pozzi, che ha saggiato il mio Inglese arrabattato e, forse, ubriaco, a Gotheborg. Gran bei tempi! Come si faceva una volta…

Grazie a Delli! Colpevole, e non solo spettatore, del mio inglese arrabattato e, beh si, ubriaco a Gotheborg. (se non vi ricordate di quale momento sto parlando, beh meglio così). Mi piace pensare di essere molto simili e ti auguro grandi soddisfazioni come quelle che, fortunatamente, mi sento di vivere ora. Ma si può e si deve sempre migliorare! Ma perché corto!?

Grazie ai cari Tova e Tex. Ho un grande affetto per voi ed il mio rammarico è di non essere un po' di più del gruppo delle celluline, cosicchè ci si sarebbe conosciuti ancora meglio! E così avremmo passato più tempo nella mitica 500, ne sono sicuro!

Grazie a tutti i cari amici ai quali con affetto, domando scusa per aver tralasciato un commento diretto a voi: in particolare saluto Laura, Michi, Rudy, Mirko, Linda, Claudia, Anna, Silvia, Manu, Cim, Cleme, Pietro, Vale, Davide e anche Salvatore.

Dopo gli affetti più di pancia, ecco i ringraziamenti alle persone che, in ambito lavorativo, più mi hanno formato:

Grazie certamente a Raffaele, che per la prima volta mi permetto di chiamare così. Professore di grande intelligenza, eccellente non solo per la pura conoscenza, ma soprattutto – quel che più conta- nel saperla trasmettere sempre con passione e professionalità al tempo stesso. Forse il mio professore preferito! ..In realtà, in estrema competizione con altri due: Garattoni e Musazzi. ...quanto tempo è passato!

Grazie di cuore a Chiara! Con la scusa di essere, se non vado errato, il tuo primo "vero" tesista, spero mi ricorderai sempre. Grazie per tutto l'aiuto, l'enorme disponibilità (direi per le notti in lab anche se, per fortuna, è stata una sola, ed in Mangiagalli), e per la grande pazienza nello spiegarmi cose che, soprattutto al'inizio, erano a dire il poco aliene a me. Grazie infinte per le ore di sonno o relax perse, soprattutto nelle ultime sere, nel correggere le mie imprecisioni. Ti auguro un gran bene e felicità. Te lo meriti!

Grazie ad Emanuela; i primi momenti mi hai seguito più di Chiara e ho imparato davvero tanto. Mi hai insegnato cose che ancora oggi non mi vengono naturali ma spero lo diventino a breve. Anche tu mi hai dedicato tantissimo tempo in queste ultime battute. Grazie per tutte le consulenze ed i consigli. Ti auguro con tutto il cuore di esaudire il tuo sogno e di tagliare questo nuovo grande traguardo che ti sei prefissata.

Grazie a tutti voi del TBM: Grazie a Dario che con la tua simpatia fai sembrare facile qualsiasi cosa, e riempi il lavoro di attimi divertenti e belli da ricordare; grazie a te Barbara, la prima con cui ho affrontato un nuovo modo di lavorare/studiare che è quello più vicino alla collaborazione cui ti spinge il lavoro. Spero tu sia stata orgogliosa dell'impegno che ho dato. Grazie ad Ilaria e Leonardo, capaci sin da subito di farmi ambientare al meglio in ufficio. Grazie!

Grazie anche a Matteo, che con enorme disponibilità mi ha permesso di sviluppare, in tutti i sensi, una importante parte della tesi.

Insomma, GRAZIE a TUTTI!! TUTTI DAVVERO!

Perché ogni persona conosciuta mi ha portato qui,

e sono orgoglioso di esserci arrivato!

Con il rischio di avervi tediato, ma con la speranza di avervi allietato, ho scritto forse più del solito.
"Ma se in vece fossimo riusciti ad annoiarvi, credete che non s'è fatto apposta."