

# Using Time-Oriented Data Abstraction Methods to Optimize Oxygen Supply for Neonates

Andreas Seyfang<sup>1</sup>, Silvia Miksch<sup>1</sup>, Werner Horn<sup>2</sup>, Michael S. Urschitz<sup>3,4</sup>,  
Christian Popow<sup>4</sup>, Christian F. Poets<sup>3</sup>

<sup>1</sup> Institute of Software Technology, University of Technology, Vienna, Austria  
`seyfang@ifs.tuwien.ac.at`

<sup>2</sup> Department of Medical Cybernetics and Artificial Intelligence, University of Vienna, and Austrian Research Institute for Artificial Intelligence, Austria

<sup>3</sup> Department of Neonatology and Pediatric Pulmonology, School of Medicine, Hannover, Germany

<sup>4</sup> Department of Pediatrics, University of Vienna, Austria

**Abstract.** Therapy management needs sophisticated patient monitoring and therapy planning, especially in high-frequency domains, like Neonatal Intensive Care Units (NICUs), where complex data sets are collected every second. An elegant method to tackle this problem is the use of time-oriented, skeletal plans. *Asgaard* is a framework for the representation, visualization, and execution of such plans. These plans work on qualitative abstracted time-oriented data which closely resemble the concepts used by experienced clinicians.

This paper presents the data abstraction unit of the *Asgaard* system. It provides a range of connectable data abstraction methods bridging the gap between the raw data collected by monitoring devices and the abstract concepts used in therapeutic plans. The usability of this data abstraction unit is demonstrated by the implementation of a controller for the automated optimization of the fraction of inspired oxygen (FiO<sub>2</sub>). The use of the time-oriented data abstraction methods results in safe and smooth adjustment actions of our controller in a neonatal care setting.

## 1 Introduction

Most Intensive Care Units (ICUs) are well equipped with modern devices for patient monitoring. On-line recording of patient data and storage in computer-based patient records (CPR) and patient data management systems (PDMS) become common place in today's ICUs. Still, the workload of the clinical staff prevents the optimal utilization of these measurements since it limits the number of adjustments during daily routine. To ease a particular part of that workload we have developed a system that helps the medical staff in automatically providing the necessary oxygen supply to newborn infants within the optimal range.

Previous experience [6, 10] shows that patient monitoring may be improved by using therapy management strategies, e.g., by using a data abstraction unit for processing the raw monitoring data. This unit must be integrated into a larger

system that supports therapy planning and the execution of clinical protocols. In this paper we describe temporal data-abstraction methods and their application to control the adjustment of the fraction of inspired oxygen ( $\text{FiO}_2$ ) based on continuously recorded pulseoximeter derived arterial oxygen saturation ( $\text{SpO}_2$ ) monitoring data. The system is part of the Asgaard framework [9] which supports the application of time-oriented, skeletal plans in the medical domain.

### 1.1 $\text{FiO}_2$ Controller

The oxygen delivery to a premature or sick newborn infant must be adjusted very closely in order to grant adequate tissue oxygenation while minimizing possible toxic effects of supplying oxygen.

Correct  $\text{FiO}_2$  settings are achieved by manual adjusting the  $\text{FiO}_2$  according to the oxygen saturation readings from pulseoximetry, transcutaneous and invasive blood gas measurements. In some patients this requires frequent adjustments within short intervals. In order to reduce the cumbersome frequent alarms and  $\text{FiO}_2$  adjustments, we often notice generous  $\text{SpO}_2$  limits resulting in an oversupply of oxygen. We therefore consider manual adjustments of the  $\text{FiO}_2$  not as an optimal solution.

The aim of our project is to develop a continuously operating automated  $\text{FiO}_2$  controller for optimizing the oxygen delivery to newborn infants. The automated  $\text{FiO}_2$  controller will adjust the  $\text{FiO}_2$  settings based on the continuous transcutaneous  $\text{SpO}_2$  measurements. During each control cycle, data are read from the pulseoximeter, validated, and in case preset  $\text{SpO}_2$  limits are exceeded or fall short of, the  $\text{FiO}_2$  is adjusted accordingly.

A similar approach is the fuzzy logic assisted controller developed by Sun et al. [11]. However, the functionality of this fuzzy controller has the important disadvantage that it resulted in an huge number of  $\text{FiO}_2$  changes. This could lead to disturbances of the immature respiratory system of preterm infants and should be avoided.

The controller described in this paper uses temporal data abstraction. It accounts for stability of the data, recognizes trend, and applies qualitative reasoning similar to that performed by clinicians.

### 1.2 Temporal Data Abstraction

Data abstraction derives meaningful information (seen as abstractions) from raw data able to support specific knowledge-based problem-solving activities. Temporal data abstraction methods represent an important subgroup where the processed data are temporal [5].

Common methods of intelligent data analysis are time-series analysis, control theory, and probabilistic or fuzzy classifiers [2]. These approaches have a lot of shortcomings, which lead to applying knowledge-based techniques to derive qualitative values or patterns of current and past situations of a patient. The RÉSUMÉ project [10] performs temporal abstraction of time-stamped data

without predefined trends. The system is based on a knowledge-based temporal-abstraction method. Larizza et al. [4] have developed methods to detect predefined courses in a time series. Complex abstraction allows one to detect specific temporal relationships between intervals. Bellazzi et al. [1] utilize Bayesian techniques to extract overall trends from cyclic data in the field of diabetes.

All these approaches are dealing mostly with low-frequency data, i.e., only few data items per day have to be processed. On the contrary, in the ICU we receive a set of data items every second. This results in the problems of oscillating data, frequently shifting contexts, repeating patterns of states and events as well as different expectations of the development of parameters.

Promising approaches for high-frequency data are the Time Series Workbench [3] which approximates data curves through a series of line-segments and VIE-VENT [6]. The temporal abstraction module of VIE-VENT focuses on the high-frequency domain of artificial ventilation of newborn infants. VIE-VENT incorporates fixed, but context-dependent temporal data abstraction schemata. It uses an epsilon region to reduce frequent qualitative changes when oscillations occur around borders of qualitative regions. However, none of these two methods offer the flexibility of a general data abstraction unit which provides methods for stable processing even on changing signal quality.

To overcome limitations observed in previous approaches, namely to arrive at a system combining temporal data abstraction with the execution of skeletal, time-oriented plans representing clinical treatment plans, the *Asgaard* framework was developed.

In the following section we describe the data abstraction methods of the *Asgaard* system. Their use in a practical application—the  $\text{FiO}_2$  controller—is demonstrated in Section 3. This is followed by notes on the evaluation and concluding remarks.

## 2 Data Abstraction in *Asgaard*

In this section we first give an overview of those parts of *Asgaard* dealing with plan execution and data abstraction followed by a description of the data abstraction unit in greater detail.

### 2.1 Introduction to *Asgaard*

The *Asgaard* framework [9] outlines task-specific problem-solving methods to support both design and execution of skeletal plans. This project tries to build a bridge between the medical approaches and the planning approaches, addressing the demands of the medical staff on the one side and applying rich plan management on the other side. For the representation of plans, a time-oriented, intention-based, skeletal plan-representation language, called *Asbru*, was developed [8]. *Asbru* is used to define skeletal plans which are instantiated during plan execution. Figure 1 shows those parts of the *Asgaard* framework which deal with the execution of plans including the data abstraction unit:

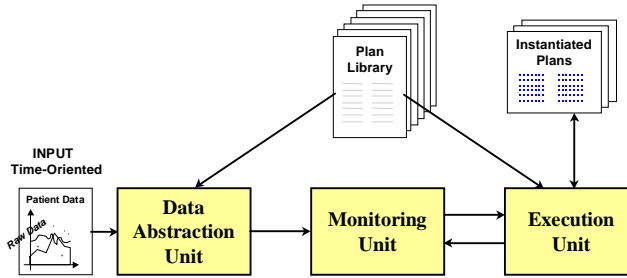


Fig. 1. Runtime Modules of Asgaard.

- Data Abstraction Unit: The data abstraction unit receives the data from the input sources and processes it through a set of abstraction methods described in the next section.
- Monitoring Unit: The monitoring unit receives a list of parameter propositions (i.e., parts of conditions) which are relevant for future plan-state transitions from the execution unit and compares it to the high-level abstractions delivered by the data abstraction unit. If it detects a match, it informs the execution unit which performs the resulting state transitions.
- Execution Unit: The execution unit handles the state transitions of the plans. It instantiates plans from the plan library and governs their life-cycle according to the state of the world as reported by the monitoring unit.

## 2.2 The Data Abstraction Unit

Asgaard implements 15 classes of abstraction methods which can be connected to each other on instantiation. Their instances are configured in the domain definition of the plan library. We distinguish the following classes (written in *italics*).

- A *raw data interface* performs error checking on data directly coming from the input sources.
- *Numeric calculations* are based on parameters and constants and comprise elementary arithmetic functions.
- *Logical combinations* are applied on Boolean parameters. Like the other abstractions, they can cope with undefined values.
- The *spread* [7] is used to smooth noisy raw data. We slide a time window of constant width over the curve in small steps. For each position of the time window, we calculate a linear regression of the valid data points within the window. On the center of the line we plot the adapted standard error. Connecting the ends of each bar with those of its neighbors yields a band (called *spread*). It vertically follows the average of the curve. The width shows the uncertainty involved in its calculation. The smaller the spread, the better the quality of the curve.

- Based on the spread, a set of abstractions are derived: *slope*, *standard deviation*, *standard error*, *center*, and *end point* (of the regression line), and *time to alarm* (intersection of the elongation of the regression line with a defined threshold value).
- Quantitative abstractions based on raw data are *change rate* and *average*. In contrast to the spread, they are calculated in the conventional way which is more suitable for low-frequency data.
- *Qualitative abstractions* are based on context-sensitive transformations of quantitative parameters using context-sensitive transformation schemata based on the spread.
- *Logical dependency* definitions yield qualitative or quantitative abstractions based on a set of logical expressions.
- *Boolean parameters* are abstracted from logical expressions.

In practical applications, instances of the above classes are connected in a directed graph which fills the gap between the input sources and the monitoring unit. Each instance receives time-stamped data (one measurement at a time) from more basic instances of abstraction methods, refines the information to a higher level of abstraction and passes the result to other instances along the graph.

Both the connections between the abstraction methods and their parameters are defined in the domain definition of the Asbru plan-library.

### 3 The FiO<sub>2</sub> Controller – A Practical Example

In the following we describe an application of Asgaard’s data abstraction capabilities to control the oxygen delivery to premature newborn infants. The aim of our controller is the optimization of oxygen supply through small adjustments at a moderate rate. It is designed for closed-loop operation which adjusts the FiO<sub>2</sub> by executing instantiated skeletal plans. For safety considerations, changes exceeding a limit of 10% are to be acknowledged by the medical staff. Adjustments result from an abstract view of the SpO<sub>2</sub> data stream.

#### 3.1 Temporal Data Abstraction for the FiO<sub>2</sub> Controller

In this section we show how the instances of Asgaard’s data abstraction methods are configured to form the controller necessary to meet the aims described above. The overall goal is to keep the SpO<sub>2</sub> in the range of normal values with minimum adjustments. We distinguish four modes of operation:

Normal operation: If none of the following exceptions described below occur, the adjustment based on the current abstractions from the SpO<sub>2</sub> readings is performed and wait mode is entered. If the SpO<sub>2</sub> is in the target region, no adjustment is performed.

Wait mode: After any change in the  $\text{FiO}_2$  setting the system waits for five minutes (This parameter depends on the mode of ventilation.) before it takes another action based on the fact that it takes that long until the change in the  $\text{FiO}_2$  setting shows an effect in the  $\text{SpO}_2$  of the patient. This is implemented in the filter conditions of the plans [8] performing the adjustment. These plan details are not described in this paper.

Check mode: Since the signal from pulseoximetry is not valid all the time, a set of criteria is defined under which the system temporarily suspends its actions. They are detailed below. The system displays the check mode but no acknowledgment by the user is demanded since our system is not aimed at alarming but at optimization of the  $\text{FiO}_2$  in standard situations.

Postpone mode: If the system should adjust the  $\text{FiO}_2$ , but most recent information seems to invalidate this action the adjustment is postponed until the short term observations match the intended action or another adjustment is considered which is in accordance with recent observations. The reasoning behind this is given below.

Figure 2 shows the instances of abstraction methods employed in our example and the connections between them. Figure 3 shows a retrospective analysis of a 45-minutes sequence of  $\text{SpO}_2$  recording together with the output of the controller.

In the following we describe the instances of the abstraction methods used to form the  $\text{FiO}_2$  controller.

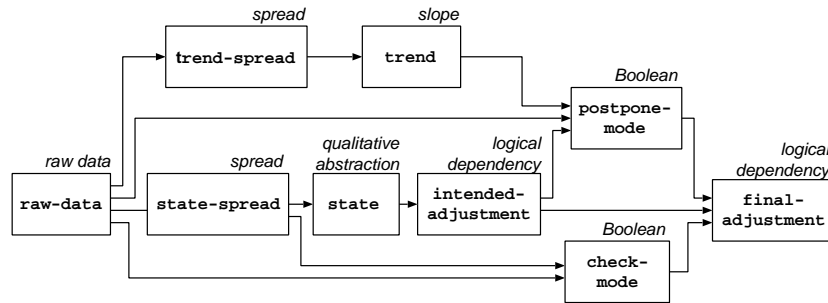
**Raw Data Selection and Validation.** The interface to the data delivered by the pulseoximeter is defined in the specification of `raw-data`. It contains the low-level identification of the parameter  $\text{SpO}_2$  and the range of values allowed.

**Underlying Spreads.** Two spreads [7] are fundamental for the abstraction process: the `state-spread` and the `trend-spread`. The `state-spread` is calculated from a time-window of five minutes as a basis of the qualitative `state` of the  $\text{SpO}_2$ , while the `trend-spread` is based on a one-minute time-window to include most recent development in the  $\text{SpO}_2$  reading.

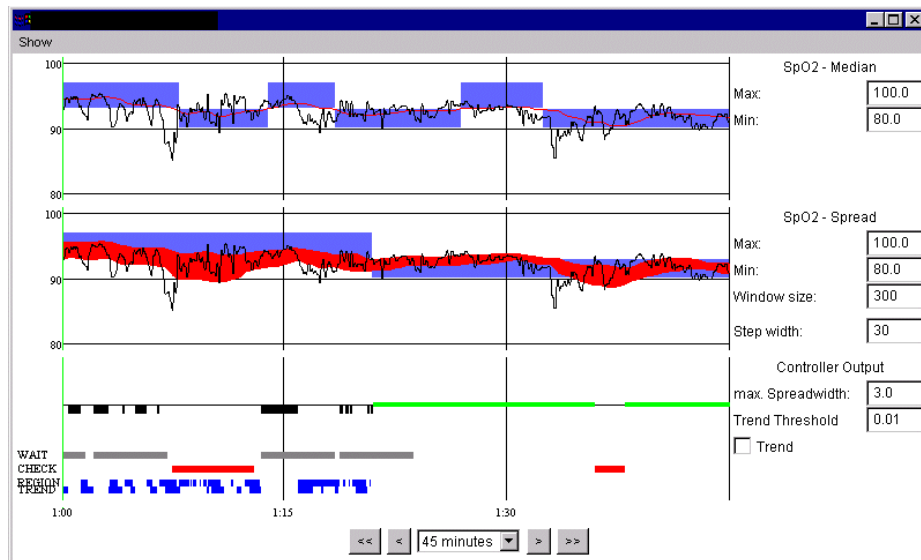
**State and Trend.** On the *center* of the `state-spread` the *qualitative abstraction state* of the  $\text{SpO}_2$  is obtained. Five qualitative values are derived from the state: substantially above, above, normal range, below, and substantially below. They are the foundation of the adjustment values for the  $\text{FiO}_2$ . Based on the *slope* of the `trend-spread` the `trend` is abstracted. It knows whether  $\text{SpO}_2$  is increasing, stable, or decreasing.

**Check Mode.** The credibility of a system heavily depends on its ability to recognize its limits. We therefore implemented in `check-mode` (an instance of *Boolean*) the following conditions under which the system signals that the situation is beyond its scope of operation:

- If the *standard error* of the `state-spread` calculated for the previous five minutes exceeds a limit of 1.5, data are too unreliable to be a basis of further reasoning. Figure 3 shows two such periods. These are visualized by red/gray bars in line CHECK.



**Fig. 2.** Combination of abstraction methods used in the  $\text{FiO}_2$  controller. The names of the **instances** of the abstraction methods are inside the boxes while their classes are given in *italics* on top of them.



**Fig. 3.** Retrospective analysis of controller operation using recorded data. The top graph shows the qualitative values abstracted from the median of the  $\text{SpO}_2$ . The graph in the middle shows the **state-spread** and the qualitative values (**state**, displayed as horizontal bars) abstracted from it. Below we display the controller output (from top to bottom): The **intended-adjustment**: first decrease  $\text{FiO}_2$  (in black) multiple times until 1:20, then the target region is reached (green/light gray bars); the periods of wait mode following each adjustment (gray); two periods of check mode (red/dark gray); and a series of short periods during which the value of **raw-data** contradicts the **intended-adjustment** (line REGION) or the **trend** dissuades from decreasing  $\text{FiO}_2$  (line TREND), both in blue/black.

- If the input from the pulsoximeter (**raw-data**) is missing or invalid due to movements of the patient or the sensor, any change of the  $\text{FiO}_2$  would not seem plausible, even if 95% of the data in the time window under consideration were valid.
- If the current reading of  $\text{SpO}_2$  drops below 80% for more than four seconds, an acute hypoxia is occurring which will be signaled by the pulsoximeter. Since our system is aimed at the optimization of  $\text{SpO}_2$  in normal situations, it cannot contribute to handling such emergencies.

**Postpone Mode.** In addition to the above reasons for pausing operation there are situations during normal operation in which it is not feasible to take the intended action since the current data contradicts the results of the analysis of the previous minutes. In the following situations the adjustment is postponed:

- If the trend observed for the last minute shows a change in the same direction which the adjustment tries to achieve, an additional change in the  $\text{FiO}_2$  setting could lead to over-reactions. E.g., if the past five minutes suggest decreasing the oxygen supply but during the last minute  $\text{SpO}_2$  was decreasing without such intervention, further observation is necessary. If the trend continues, the desired effect takes place without the need for adjustments. If the trend changes unfavorably, the  $\text{FiO}_2$  setting must be changed according to the pending adjustment. The left half of Figure 3 shows a series of short intervals (as blue/black bars in line **TREND**) during which the graph of the raw data descends, rendering a reduction of  $\text{FiO}_2$  superfluous.
- If the most recent  $\text{SpO}_2$  reading grossly contradicts the intended adjustment, the latter is not issued. This is the case if the system intends to decrease the  $\text{FiO}_2$  and the current reading is normal or even below normal or vice versa. Again we postpone the action. Line **REGION** in Figure 3 shows these periods.

**Intended and Final Adjustment.** Based on the **state**, an **intended-adjustment** is calculated. If neither **check-mode** nor **postpone-mode** are active (represented as true-value of these parameters), the **final-adjustment** takes the value of the **intended-adjustment**. Otherwise no adjustment is recommended, i.e., **final-adjustment** is zero.

Figure 3 shows a period from 1:00 to 1:20 during which decreasing the  $\text{FiO}_2$  by 1% is intended, interleaved with one period of **check-mode** and a series of short periods during which either the **trend** or a mismatch of **state** and **raw-data** hint at postponing the adjustment. From 1:20 onwards the qualitative value of  $\text{SpO}_2$  is in the target region and no adjustment is necessary as marked by the green/gray bar.

The output of the data abstraction—**final-adjustment**—is fed into the plan execution unit via the monitoring unit which performs the instantiation and execution of the plans (compare Figure 1). The plans refer to the **final-adjustment** in their filter conditions, i.e., an appropriate user-performed plan is started to realize the adjustment whenever necessary.



## 4 Limitations

The spread algorithm is suitable for high-frequency, noisy signals. It is not appropriate for low-frequency data nor for data which shows meaningful oscillations. However, the data-abstraction unit can handle low-frequency data using other modules.

The current implementation handles several measurements per second without problems. The absolute limit of performance depends on the complexity of the instantiated data-abstraction, the performance of the computer used, and the data format and transmission speed of the measuring device(s) connected to the serial port.

## 5 Evaluation and Further Work

We currently perform a three step evaluation process of the  $\text{FiO}_2$  controller. In a first step, we compared the spread to median filtering. The spread leads to a more stable judgment of the patient's situation and therefore reduces the number of unnecessary adjustments of  $\text{FiO}_2$  (see Fig. 3, top and middle graphs). We retrospectively evaluated recordings obtained from 10 patients. In a total of 126 hours (median 12, range 3–24) the adjustment based on the *spread* changed 148 times compared to 519 changes in the adjustment based on the median.

In a second step, we implemented the logic described above to derive adjustments of the  $\text{FiO}_2$  setting. This proved to fully eliminate controversial adjustments.

In the next step the system will be evaluated online in a Neonatal Intensive Care Unit (NICU). The clinical study is designed to incorporate 13 premature infants, first 3 for tuning the parameters of the controller, the others to evaluate the performance. For safety reasons the system will run in open-loop mode. The necessary adjustments to the  $\text{FiO}_2$  are displayed to the medical staff and set manually. In three consecutive 2-hours periods the neonates are managed by the ward staff, an expert giving special attention to ventilation management, and the open-loop controller. Comparison of performance will give insights to improvements in health-care offered by automated devices.

## 6 Conclusion

The work described in this paper shows that temporal data abstraction needs a complex intelligent toolset to provide complex solutions necessary in real-world applications such as optimizing the oxygen supply of premature neonates. The data abstraction facilities provided by the Asgaard system are powerful enough to implement such a system.

The design using formal data abstraction methods offers the flexibility to adapt the  $\text{FiO}_2$  controller to different settings required by the various ventilation modes used in the NICU. The controller is flexible enough to be used with an incubator, in CPAP (continuous positive airways pressure) mode and with

controlled ventilation. Future work will be devoted to fine-tune the parameters to various situations in the field of artificial ventilation of neonates.

**Acknowledgments.** This project is supported by "Fonds zur Förderung der wissenschaftlichen Forschung - FWF" (Austrian Science Foundation), P12797-INF. We appreciate the support given to the Austrian Research Institute of Artificial Intelligence (ÖFAI) by the Austrian Federal Ministry of Education, Science, and Culture.

## References

1. R. Bellazzi, C. Larizza, P. Magni, S. Montani, and G. De Nicolao, 'Intelligent analysis of clinical time series by combining structural filtering and temporal abstractions', in *Artificial Intelligence in Medicine*, ed., Horn, W. et al., pp. 261–270, Berlin, (1999). Springer.
2. M. Berthold and D.J. Hand, *Intelligent Data Analysis: An Introduction*, Springer, Berlin, 1999.
3. J. Hunter and N. McIntosh, 'Knowledge-based event detection in complex time series data', in *Artificial Intelligence in Medicine*, ed., Horn, W. et al., pp. 271–280, Berlin, (1999). Springer.
4. C. Larizza, R. Bellazzi, and A. Riva, 'Temporal abstractions for diabetic patients management', in *Artificial Intelligence in Medicine*, ed., Keravnou E. et al., pp. 319–330, Berlin, (1997). Springer.
5. N. Lavrac, E. Keravnou, and B. Zupan, 'Intelligent data analysis in medicine', in *Encyclopaedia of Computer Science and Technology*, ed., Kent, A. et al., volume 42, 113–157, Marcel Dekker, New York, Basel, (2000).
6. S. Miksch, W. Horn, C. Popow, and F. Paky, 'Utilizing temporal data abstraction for data validation and therapy planning for artificially ventilated newborn infants', *Artificial Intelligence in Medicine*, **8**(6), 543–576, (1996).
7. S. Miksch, A. Seyfang, W. Horn, and C. Popow, 'Abstracting steady qualitative descriptions over time from noisy, high-frequency data', in *Artificial Intelligence in Medicine*, ed., Horn, W. et al., pp. 281–290, Berlin, (1999). Springer.
8. A. Seyfang, R. Kosara, and S. Miksch, 'Asbru 7.2 reference manual', Technical Report Asgaard-TR-2000-3, Vienna University of Technology, Institute of Software Technology, (2000). available at <http://www.ifs.tuwien.ac.at/asgaard/asbru/asbru.7.2.new/>.
9. Y. Shahar, S. Miksch, and P. Johnson, 'The Asgaard Project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines', *Artificial Intelligence in Medicine*, **14**, 29–51, (1998).
10. Y. Shahar and M. A. Musen, 'Knowledge-based temporal abstraction in clinical domains', *Artificial Intelligence in Medicine*, **8**(3), 267–298, (1996).
11. Y. Sun, I.S. Kohane, and A.R. Stark, 'Computer-assisted adjustment of inspired oxygen concentration improves control of oxygen saturation in newborn infants requiring mechanical ventilation', *Journal of Pediatrics*, **131**(5), 754–756, (1997).