Using Model Checking to Detect Masking in IEC 60601-1-8-compliant Alarm Configurations

Matthew L. Bolton
University at Buffalo, SUNY

Bassam Hasanain
University of Illinois at Chicago

Andrew D. Boyde
University of Illinois at Chicago

Judy Edworthy
Plymouth University

The failure of humans to respond to auditory medical alarms has resulted in numerous patient injuries and deaths. The widely used IEC 60601-1-8 international medical alarm standard was created to improve alarm discernibility and identification. Unfortunately, the melodic tonal patterns of IEC 60601-1-8's alarms are particularly susceptible to simultaneous masking, a condition where concurrent sounds interact in ways that make one or more of them imperceptible. This paper presents a method, which builds on a previous implementation, that uses a novel combination of psychophysical modeling and model checking to detect masking in a modeled configuration of IEC 60601-1-8 alarms. We describe our updated method and demonstrate its power by using it to find masking in the alarms of an actual IEC 60601-1-8-compliant telemetry monitoring system. Results and future research are discussed.

INTRODUCTION

Auditory medical alarms have many problems that can make them difficult to perceive (Edworthy, 2013). The Pennsylvania Patient Safety Authority reports that there have been 194 documented problems with operators' responses to telemetry monitoring alerts from June 2004 to December 2008 resulting in at least 12 deaths (ECRI Institute & ISMP, 2009). A Sentinel Event Alert issued in 2013 reported 98 alarm-related incidents: 80 resulted in patient death, 13 in "permanent loss of function," and 5 extended patient hospital stays (The Joint Commission, April 8, 2013). These problems occur because the number of alarms sounding in modern medical environments often exceeds human perceptual and cognitive capabilities (Edworthy, 2013; The Joint Commission, April 8, 2013).

The simultaneous masking of alarms is an understudied issue that occurs when multiple alarm sounds interact in a way that prevents the human sensory and perceptual systems from hearing one of or more of them (Fastl & Zwicker, 2006). This is a very real and serious problem that has been acknowledged by many experts and researchers (Edworthy & Meredith, 1994; Patterson & Mayfield, 1990) and detected in clinical settings (Momtahan, Hetu, & Tansley, 1993; Toor, Ryan, & Richard, 2008). However, the vast majority of the work on alarm safety has focused on other areas (Edworthy, 2013). Auditory masking can be very difficult to detect because it may only occur with very specific interactions of multiple, concurrently sounding medical alarms (Hasanain, Boyd, & Bolton, 2015).

The IEC 60601-1-8 international standard (2003) was created to give engineers guidance about how to design and test medical alarms so that they are "readily discernible without being unnecessarily distracting or disturbing." To accomplish this, the standard contains a set of reserved alarm sounds (for common alarm conditions) and instructions for creating additional alarms. The melodic patterns of tones that are specified within IEC 60601-1-8 make them particularly susceptible to simultaneous masking (Fastl & Zwicker, 2006). This is a very dangerous situation because engineers relying on the IEC 60601-1-8 standard may be designing alarms that are facilitating masking in medical environments. Further, as the number of alarms in medicine increases, this is a problem that will get worse.

In the work presented here, we introduce a method (itself an extension of previous work (Hasanain et al., 2015)) capable of proving whether or not masking can manifest in a modeled configuration of tonal medical alarms, like those in IEC 60601-1-8. Our method uses the psychoacoustics of simultaneous masking (mathematical formulas that can determine if masking occurs between sounds) synergistically with model checking (a computational tool that automatically proves properties about state-machine-based models). In this paper we present our method and show how it can be used to detect masking in IEC 60601-1-8-compliant alarm designs.

BACKGROUND

In the following, we cover the necessary background on model checking, the psychoacoustics of simultaneous masking, and the previous version of our method.

Model Checking

Model checking is an automated approach to formal verification (Clarke, Grumberg, & Peled, 1999). A model describes a system as a state machine. Temporal logic specification properties assert desirable model conditions using temporal logic. Verification processes exhaustively search through the system model to prove whether or not the properties hold. If they do, the model checker returns a confirmation. If there is a violation, an execution trace called a counterexample is produced that shows how the failure occurred. Model checking is particularly good at finding problems in systems with concurrency, where system elements can interact in unanticipated ways.

Researchers have used model checking to successfully find and correct human factors issues in automated systems (Bolton, Bass, & Siminiceanu, 2013). Outside of our previous results (Hasanain et al., 2015), none of this work has explored how human perceptual issues can be evaluated with formal analyses.

The Psychoacoustics of Simultaneous Masking

The psychoacoustics of simultaneous masking mathematically relate a sound's physical characteristics (its frequency/tone and volume) to the masking effect the sound has on human perception. The most successful of these are based on the expected excitation patterns of the human ear's basilar membrane (the physical structure largely responsible for allowing humans to distinguish between different sounds). Conceptually, these models predict how a potentially masking sound (the *masker*) will stimulate the receptors on the basilar membrane

Copyright 2016 by Human Factors and Ergonomics Society. DOI 10.1177/1541931213601146

based on its volume and relative frequency to the potentially masked sound (the *maskee*). This stimulation creates a higher volume threshold (in dB) that the *maskee* must exceed to be perceivable (Bosi & Goldberg, 2003).

A masking threshold is represented by a "masking curve":

$$\text{curve}_{masker}(z_{maskee}) = \text{spread}_{masker}(\delta z) + v_{masker} - \Delta. \quad (1)$$

v_{masker} is the volume of the masker in dB. δz is defined as

$$\delta z = z_{maskee} - z_{masker}, \quad (2)$$

where z_{maskee} and z_{masker} are the frequency of the maskee and masker respectively on the Bark scale (the Bark scale maps a frequency in Hx to a location on the ear's basilar membrane where the sound stimulates the receptors the strongest.). spread_{masker} defines how the volume of the masking threshold changes with δz. Δ is the minimum difference between a masker's and maskee's volume under which masking can occur. There are many psychoacoustic spreading functions and Δs for different types of sounds (Bosi & Goldberg, 2003).

These psychoacoustics can determine if a single sound can mask another and were the basis for previously works (Hasanain et al., 2015). However, when there are multiple concurrent sounds, their combined masking threshold can be greater than the sum of each individual masker's effect. This additive masking (Bosi & Goldberg, 2003) is modeled by combining the masking curve values of each potential masker on the power scale. Using the following equation to represent a volume (v in dB) on the power scale

$$\text{power}(v) = 10^{v/10}, \quad (3)$$

for a given potential maskee and N potential maskers, the aggregate masking threshold (in dB) is calculated as

$$\begin{aligned}\text{power}(\text{mthresh}_{maskee}) &= \text{power}(\text{abs}_{maskee}) \\ &+ \left(\textstyle\sum_{n=1}^{N} \text{power}\left(\text{curve}_{masker_n}(z_{maskee})\right)^{\alpha}\right)^{1/\alpha}.\end{aligned} \quad (4)$$

In this, α is a positive constant (Green, 1967) and abs_{maskee} is the absolute threshold of hearing (in dB) at the maskee's frequency (f_{maskee} in Hz) calculated as (Terhardt, 1979)

$$\begin{aligned}\text{abs}_{maskee} &= 3.64 \cdot \left(\frac{f_{maskee}}{1000}\right)^{-0.8} \\ &- 6.5 \cdot e^{-0.6\left(\frac{f_{maskee}}{1000} - 3.3\right)^2} + 10^{-3} \cdot \left(\frac{f_{maskee}}{1000}\right)^4.\end{aligned} \quad (5)$$

These psychoacoustics been used to predicting masking for normal human hearing for decades (Bosi & Goldberg, 2003). They have been used to identify masking between recorded medical sounds (Toor et al., 2008). They have also served as the basis for audio compression techniques like those used in MPEG (Bosi & Goldberg, 2003).

Our Original Method

In our original method (Hasanain et al., 2015), an analyst manually modeled a configuration of medical alarms based on a set architecture and code patterns. Specifications were created using patterns that asserted the absence of partial and total masking. Model checking could then be used to determine if any given alarm could ever be masked by other alarms based on the psychoacoustics in (1). It is important to note that masking could only be detected between pairs of alarms, though multiple pairs of alarms could contribute to the masking of a given alarm.

In this version of the method, we used the spreading function (for computations using (1)) from the MPEG2 audio codec (Schroeder, Atal, & Hall, 1979). We also used

$$\Delta = 14.5 + z_{masker}. \quad (6)$$

While this version of the method proved itself to be useful (Hasanain et al., 2015) it had several limitations. Because it only considers masking between pairs or alarm sounds, it does not account for the additive effect of masking (Bosi & Goldberg, 2003). Further, the used spreading function and Δ were used because they facilitated a computationally efficient implementation. Thus, there are more appropriate psychoacoustics. The new method we present here addresses these limitations.

THE NEW METHOD

The work presented here shows how our method was reimplemented to address the limitations of the original. In this new version, we enable our method to account for additive of masking. We also update the psychoacoustics used to compute masking curves to better reflect the tonal nature of the masking sounds of alarms. The updated version of our method is shown in Figure 1. An analyst first examines alarm documentation and describes the behavior of the alarms using a spreadsheet, where each alarm is described as a sequences of tones (and pauses between tones) each with a defined frequency (Hz), volume (dB), and duration (s). The analyst uses a computer program to automatically convert the described alarm configuration into a formal model and specification properties that assert the absence of masking. Model checking (in our case the Symbolic Analysis Laboratory (SAL); De Moura et al. 2004) is then used to prove whether the model will always satisfy the properties. If it does, the verification report indicates that they were proved. If a specification is violated, the returned counterexample shows exactly how an alarm was masked. Below describes the formal model and specifications in more detail.

Formal Model

The formal model used in the method is automatically generated by our computer program and has a set architecture (Figure 2). It is made of a set of synchronously composed submodels, each with a particular purpose. All are discussed below.

The Clock. The clock sub-model is unchanged from the previous version of the method (see Hasanain et al. 2015). It uses a timed automaton (Dutertre & Sorea, 2004) to advance model time (*Time*) and communicate it to the other sub-models.

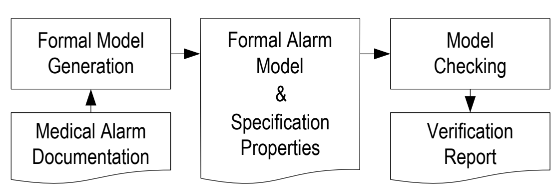

Figure 1. Masking detection method flow chart.

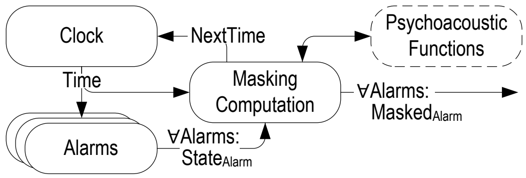

Figure 2. Alarm configuration formal modeling architecture.

The *Time* is initially 0. Then, for every following model step, *Time* is advanced to a new one that is greater than the current *Time* and less than or equal to the *NextTime*, an input from the masking computation sub-model.

Alarms. Each alarm is represented as a sub-model that can start or stop sounding at appropriate times and adjust its state based on its current state and how long it has been sounding. Specifically, each alarm model keeps track of whether it is sounding or not (the alarm is sounding if its *StartTime* > 0) and to change the alarm's state at set times relative to its *StartTime*. If the alarm is not sounding, at any *Time* > 0 the alarm can start by assigning the *Time* to the *StartTime*. Once started, an alarm will sound for a single cycle and then stop (set the *StartTime* to zero). The alarm can sound again in the future. Each alarm model must compute the amount of time the alarm has been sounding (*TimeInCycle* = *Time* - *StartTime*) and adjust the model's state appropriately. The alarm's state is treated as a finite state machine (Figure 3). Each alarm state represents separate tone or pause in the associated alarm's sounding cycle, where the amount of time an alarm can sound in a given state is determined by the given tone's duration. As such, each alarm state maps to a specific frequency and volume associated with a given tone or pause (frequency and volume can be 0) in the alarm. Each state also maps to a "next time" value, representing the time as which a change event will occur in the alarm (when the next tone or pause will start and/or when the alarm will stop sounding). These mappings are used by the masking computation sub-model to determine if masking is occurring.

Masking Computation and Psychoacoustics. At any given *Time* in the model, the masking computation sub-model is responsible for examining the state of all sounding alarms and determining if masking is occurring. To accomplish this, the method uses the psychoacoustics of simultaneous masking discussed above. To improve the detection capabilities of our

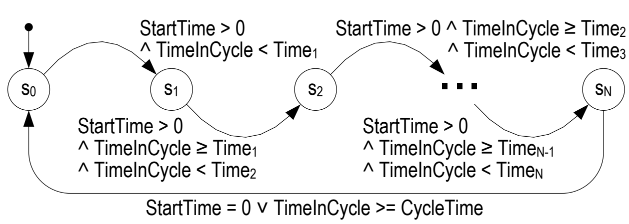

Figure 3. State machine model describing how the behavior of an alarm with N tones and/or pauses is formally modeled in our method. Circles represent states. Arrows represent transitions between states that occur when the logical conditions on the arrows are satisfied. s0 is the initial state where the alarm is not sounding. *Time* variables with subscripts represents times constant times (determined by tone durations) when an alarm should transition between states based on the *TimeInCycle*.

method, these have been updated from our previous version (Hasanain et al., 2015). When computing masking curves (1), we now use the following spreading function:

$$\text{spread}_{masker}(\delta z) = \begin{cases} -17\cdot\delta z + 0.15\cdot v_{masker} \\ \quad \cdot(\delta z - 1)\cdot\theta(\delta z - 1) & \text{for } \delta z \geq 0 \\ -(6 + 0.4\cdot v_{masker})\cdot|\delta z| \\ \; -(11 + 0.4\cdot v_{masker}\cdot(|\delta z| - 1)) & \text{otherwise} \\ \quad \cdot\theta(|\delta z| - 1) \end{cases} \tag{7}$$

where $\theta(x) = 1$ for $x \geq 0$ and $\theta(x) = 0$ otherwise. This particular spreading function was chosen because it is the most appropriate for modeling the masking effects of tones on other tones (Brandenburg & Stoll, 1994). We also updated the way that Δ (from (1)) was computed. In the new version

$$\Delta = 6.025 + 0.275\cdot z_{masker}\ \text{dB}. \tag{8}$$

This was used for several reasons. First, it has been shown to be appropriate for tones (Ambikairajah, Davis, & Wong, 1997). It was also used in the MPEG audio codec (Bosi & Goldberg, 2003), thus it has a well-established validity. Further, it will always be smaller than the Δ used in the original method (6). This means that it will increase the chances that our method will detect masking. Given that missing the detection of masking has significantly worse consequences than a false alarm, this was a preferable value of Δ for our purposes.

Further, to account for additive masking, all of the above were used with (4) with $\alpha = 0.33$, a value Lutfi (1983) found best captured the "over adding" of the masking effects of tones. This means that masking can be said to occur if

$$\begin{aligned} &\text{power}(\text{mthresh}_{maskee}) \leq \text{power}(\text{abs}_{maskee}) \\ &\quad + \left(\textstyle\sum_{n=1}^{N} \text{power}(\text{curve}_{masker_n}(z_{maskee}))^{\alpha}\right)^{1/\alpha}. \end{aligned} \tag{9}$$

All of these psychoacoustics pose a problem for model checking. Specifically, model checkers are not capable of handling the non-linear arithmetic of model variables. Thus these psychoacoustics cannot be represented in a formal model in their native form. To address this, the psychoacoustics are represented in the formal model as a pre-computed lookup table (generated by the computer program).

The premise of this lookup table is that Equation (9), though basic algebraic operations, can be represented as

$$\begin{aligned} &(\text{power}(v_{maskee}) - \text{power}(\text{absolutethreshold}_{maskee}))^{\alpha} \\ &\quad \leq \textstyle\sum_{n=1}^{N} \text{power}(\text{curve}_{masker_n}(z_{maskee}))^{\alpha}. \end{aligned} \tag{10}$$

In this, because everything to the left of the $\leq$ symbol is associated with the potential maskee; everything to the right is associated with masking curves; and masking occurs if the left is less than or equal to the right. This means that the value associated with the left side of the equation can be pre-computed and directly associated with (functionally mapped from) each state of each alarm. The values of the masking curves razed to α can then be implemented as a pre-computed lookup table, where a given value is accessed based on the alarm state of the maskee and the masker. All of these values can then be linearly added together for the comparison in (10). In this way, the masking computation sub-model is able to treat each alarm as a potential maskee and every other alarm as a potential mask and determine if the maskee will masked according to (10).

The masking computation sub-module is also responsible for computing the maximum time the clock advance to on the next step (*NextTime*; Figure 2). It does this by mapping each alarm state to a next update time (the next time the state of the alarm will change) and selecting the minimum of these values.

Specification Properties

Our computer generates specification properties using minor variations of the patterns identified in our previous work (Hasanain et al., 2015). For each alarm (the *Alarm*), two properties are created: one asserting that the alarm should never be partially masked (for all paths through the model **G** it should never be sounding and masked)

$$\mathbf{G}\neg\left(Sounding_{Alarm} \wedge Masked_{Alarm}\right) \tag{11}$$

and one that it should never be totally masked

$$\mathbf{G}\neg\left(\begin{array}{l}\neg Sounding_{Alarm} \\ \wedge\mathbf{X}\left(\begin{array}{l}Sounding_{Alarm} \wedge Masked_{Alarm} \\ \wedge\left(\left(\begin{array}{l}Sounding_{Alarm} \\ \wedge Masked_{Alarm}\end{array}\right)\mathbf{U}\left(\neg Sounding_{Alarm}\right)\right)\end{array}\right)\end{array}\right). \tag{12}$$

This can be interpreted as, for all (**G**) model paths, it should never be true that the alarm goes from not sounding, to sounding and masked in the next (**X**) state such that, from then on, the alarm is sounding and masked until (**U**) it is no longer sounding.

APPLICATION

To illustrate the capabilities of our method to analyze IEC 60601-1-8-compliant alarms, we use it to evaluate the masking potential of a realistic application based on the GE CARESCAPE^{TM}Monitor B850 (GE Healthcare, 2010), a telemetry monitoring system. The GE monitor has the alarms shown in Table 1. There were four high-priority alarms that each played the same ten-tone alarm melodies, one medium-priority alarm with a three-tone melody, and one low-priority alarm with one tone. Our analyses assumed that any of these alarms could sound simultaneously.

We modeled these alarms in our new method and formally verified each to see if they were ever partially or totally masked. Verifications were performed using SAL's infinite bounded model checker (De Moura et al., 2004) on a Linux workstation with a 3.3 GHz Intel Xeon processor and 64 GB of RAM. Our results (Table 2) show that masking is possible between the alarms of the GE CARESCAPE. Partial masking was observed for all of the alarms. Two of the alarms can be totally masked: CPU-C1 and SystemLow (Figure 4). This is concerning because it means that these alarms may not be heard or responded to by an observer. Because SystemLow is low-priority, one could argue that it being masked by higher priority alarms is not important. However, CPU-C1 is high-priority with a cycle length of 8.2 seconds. This means that if the alarm is totally masked, someone will not respond to it for at least that long. In a safety critical medical environment, this is a significant amount of time. As such, this is a major patient safety problem.

Table 1. *GE CARESCAPE Telemetry Monitor Alarms*

Names	Freq (Hz)	Time (s)	Name	Freq (Hz)	Time (s)	Name	Freq (Hz)	Time (s)
CPUC1	523	0.1	System	523	0.2	System	523	0.2
(72 dB)	0	0.1	Medium	0	0.2	Low		
	698	0.1	(83 dB)	784	0.2	(79 dB)		
D15K	0	0.1		0	0.2			
(81 dB)	784	0.1		988	0.2			
	0	0.3		0	19			
D19KT	880	0.1						
(82 dB)	0	0.1						
	988	0.1						
System	0	1						
High	523	0.1						
(84 dB)	0	0.1						
	698	0.1						
	0	0.1						
	784	0.1						
	0	0.3						
	880	0.1						
	0	0.1						
	988	0.1						
	0	5						

Note. Alarm volumes are shown in parenthesis below the alarm names. iCPU-C1, D15K, D19KT, and SystemHigh all have different volumes but have the same sounding pattern of tones. These are all high-priority alarms. SystemMedium is a medium-priority alarm. SystemLow is a low priority alarm.

DISCUSSION AND CONCLUSION

This work has introduced a novel extension of our original method. In this extension, we are now able to account for additive masking and use a more appropriate spreading function. This means that our masking prediction is more accurate and can thus detect masking conditions that it could not before. In addition, we presented a case study that demonstrates the ability of the method to detect masking in a realistic configuration IEC 60601-1-8-compliant alarms. As such, the method clearly has utility and, if used by medical device engineers and/or hospitals to evaluate and design medical alarms, could significantly increase the chance that alarms are perceivable. This could have a profound impact on patient safety.

The fact that masking was detected for all of the alarms in our IEC 60601-1-8-compliant application is concerning as it indicates that there are masking problems for alarms designed to adhere to the standard. This is potentially very dangerous. The method presented here has the potential to be used to systematically evaluate the alarm requirements in IEC 60601-1-8

Table 2. *Verification Results*

Name	Masking Spec.	Time (s)	Outcome
CPU-C1	Partial	145.70	×
	Total	60,967.05	×
D15K	Partial	135.21	×
	Total	145,870.50	✓
D19KT	Partial	135.21	×
	Total	148,252.81	✓
SystemHigh	Partial	139.02	×
	Total	395,441.48	✓
SystemMedium	Partial	104.24	×
	Total	203,702.73	✓
SystemLow	Partial	81.24	×
	Total	216.66	×

Note. A ✓ indicates a confirmation. A × indicates a counterexample.

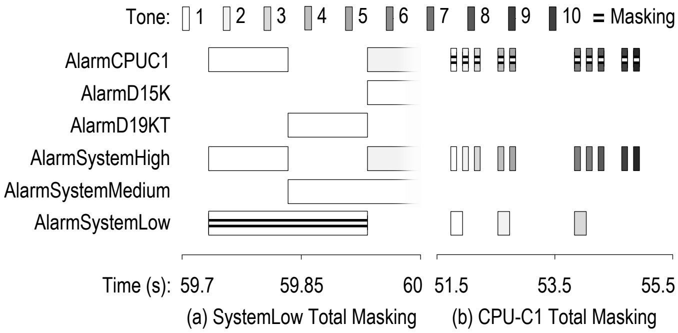

Figure 4. Visualizations of the counterexamples showing how total masking could occur for the GE CARESCAPE alarms.

and potentially explore solutions to discovered problems. For this to occur, the method will need to be extended in several ways. These are discussed below.

Scalability Improvements

Because our method uses model checking, it will scale badly (Clarke et al., 1999). Thus, scalability improvements will likely be necessary to tackle the IEC 60601-1-8 standard. It may be possible to use compositional verification (a way to verify an entire model by verifying pieces in isolation; Cobleigh, Giannakopoulou, and Păsăreanu 2003) to verify alarm configurations across multiple analyses. Further, many alarms, including those in IEC 60601-1-8, can have repeated patterns both within and between alarms. Thus, it may be possible to exploit model symmetry (Emerson & Sistla, 1996) to further improve the scalability of the method. Future efforts should investigate how these approaches scale.

More Complex Alarm Behavior and Sounds

Features of IEC 60601-1-8 are not currently supported by the method. IEC 60601-1-8 alarms can have sub-frequencies: additional simultaneous frequencies to make tones more complex. These were not considered in the presented analyses because they are not specified in the CARESCAPE documentation (GE Healthcare, 2010). Given that the method supports additive masking, accounting for these sub-frequencies should be an easy method extension in future work.

While IEC 60601-1-8 specifies tonal alarms, like the ones analyzed here, it does allow for other sounds. Further, future alarm standards will likely use more complex sounds (Edworthy, 2013). These could potentially be accommodated through the use of different spreading functions in our method (Bosi & Goldberg, 2003). This should be explored in the future.

Our current method implementation requires discrete transitions in alarm state. However, alarm sounds can have dynamic elements. Accommodating these will require significant changes to the method. Future work should investigate if there are abstraction techniques that can be used to model more dynamic types of sounds in our method.

Additional Masking Detection

Our method is capable of detecting simultaneous masking. There is also temporal masking (Fastl & Zwicker, 2006), where non-concurrent sounds can mask each other. Psychoacoustics exist for accounting for this, however these are not readily adaptable to formal modeling. Future work could investigate how to include temporal masking in our method.

REFERENCES

Ambikairajah, E., Davis, A., & Wong, W. (1997). Auditory masking and MPEG-1 audio compression. *Electronics & Communication Engineering Journal*, *9*(4), 165–175.

Bolton, M. L., Bass, E. J., & Siminiceanu, R. I. (2013). Using formal verification to evaluate human-automation interaction in safety critical systems, a review. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, *43*(3), 488–503.

Bosi, M., & Goldberg, R. E. (2003). *Introduction to digital audio coding and standards*. New York: Springer.

Brandenburg, K., & Stoll, G. (1994). ISO/MPEG-1 audio: A generic standard for coding of high-quality digital audio. *Journal of the Audio Engineering Society*, *42*(10), 780–792.

Clarke, E. M., Grumberg, O., & Peled, D. A. (1999). *Model checking*. Cambridge: MIT Press.

Cobleigh, J., Giannakopoulou, D., & Păsăreanu, C. (2003). Learning assumptions for compositional verification. *Tools and Algorithms for the Construction and Analysis of Systems*, 331–346.

De Moura, L., Owre, S., Rueß, H., Rushby, J., Shankar, N., Sorea, M., & Tiwari, A. (2004). SAL 2. In *Proceedings of the 16th international conference on computer aided verification* (pp. 496–500). Springer.

Dutertre, B., & Sorea, M. (2004). *Timed systems in SAL* (Tech. Rep. No. NASA/CR-2002-211858). SRI International.

ECRI Institute, & ISMP. (2009). Connecting remote cardiac monitoring issues with care areas. *Pennsylvania Patient Safety Authority*, *6*(3), 79–83. Retrieved from `http://patientsafetyauthority.org/ADVISORIES/AdvisoryLibrary/2009/Sep6(3)/Pages/79.aspx`

Edworthy, J. (2013). Medical audible alarms: A review. *Journal of the American Medical Informatics Association*, *20*(3), 584–589.

Edworthy, J., & Meredith, C. S. (1994). Cognitive psychology and the design of alarm sounds. *Medical Engineering & Physics*, *16*(6), 445–449.

Emerson, E. A., & Sistla, A. P. (1996). Symmetry and model checking. *Formal methods in system design*, *9*(1), 105–131.

Fastl, H., & Zwicker, E. (2006). *Psychoacoustics: Facts and models* (Vol. 22). Springer.

GE Healthcare. (2010). *CARESCAPE^{TM} monitor B850 technical specifications supplement* (Tech. Rep. No. 2040386-084D). General Electric Company.

Green, D. M. (1967). Additivity of masking. *The Journal of the Acoustical Society of America*, *41*(6), 1517–1525.

Hasanain, B., Boyd, A., & Bolton, M. (2015). Using model checking to detect simultaneous masking in medical alarms. *IEEE Transactions on Human-Machine Systems*. doi: 10.1109/THMS.2014.2379661

IEC 60601-1-8. (2003). *Medical electrical equipment - part 1-8*. Geneva: International Electrotechnical Commission.

Lutfi, R. A. (1983). Additivity of simultaneous masking. *The Journal of the Acoustical Society of America*, *73*(1), 262–267.

Momtahan, K., Hetu, R., & Tansley, B. (1993). Audibility and identification of auditory alarms in the operating room and intensive care unit. *Ergonomics*, *36*(10), 1159–1176.

Patterson, R. D., & Mayfield, T. F. (1990). Auditory warning sounds in the work environment. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, *327*(1241), 485–492.

Schroeder, M. R., Atal, B. S., & Hall, J. (1979). Optimizing digital speech coders by exploiting masking properties of the human ear. *The Journal of the Acoustical Society of America*, *66*, 1647–1652.

Terhardt, E. (1979). Calculating virtual pitch. *Hearing Research*, *1*(2), 155–182.

The Joint Commission. (April 8, 2013). Medical device alarm safety in hospitals. *Sentinel Even Alert*, *50*.

Toor, O., Ryan, T., & Richard, M. (2008). Auditory masking potential of common operating room sounds: A psychoacoustic analsysis. In *Anesthesiology* (Vol. 109, p. A1207). Park Ridge: American Society of Anesthesiologists.