Is hearing-aid signal processing ready for machine learning?

Bert de Vries[1,2,*] and Andrew Dittberner[3]

[1] *GN ReSound, Het Eeuwsel 6, 5612 AS, Eindhoven, Netherlands*

[2] *Eindhoven University of Technology, Den Dolech 2, Eindhoven, Netherlands*

[3] *GN ReSound, 2601 Patriot Blvd., Glenview, IL 60026, USA*

In the hearing-aids community, machine-learning technology enjoys a reputation as a potential performance booster for signal-processing issues such as environmental steering, personalization, algorithm optimization, and speech detection. In particular in the area of in situ hearing aid personalization, the promise is steep but clear success stories are still hard to come by. In this contribution, we analyze the 'personalizability' of typical hearing-aid signal-processing circuits. We discuss a few salient properties of a very successful adaptable and personalized signal-processing system, namely the brain, and we discover that among some other issues, the lack of a probabilistic framework for hearing-aid algorithms hinders interaction with machine-learning techniques. Finally, the discussion leads to a set of challenges for the hearing-aid research community in the quest towards in situ personalizable hearing aids.

INTRODUCTION

In this paper, we distinguish three groups of hearing-aid (HA) algorithm designers. By a designer we mean any entity that is capable to affect the input-output behavior of an HA algorithm. The first designer group entails the *professionals*: engineers, scientists, and dispensing audiologists. The professionals deal with *ex situ* design. Roughly speaking, engineers and scientists define the algorithm *structure* (i.e., the equations), whereas audiologists set the HA algorithm *parameters* during a fitting session. After a patient has been fitted and he walks away with an operational hearing aid, there still remain two entities that are capable of changing the HA algorithm under *in situ* conditions. The second designer group is the *patient* himself who can update an HA algorithm through (machine-learning-based processing of) preference feedback. For instance, patient feedback, collected through a volume-control wheel, could be used to change some gain parameters of the hearing aid. Finally, the *acoustic environment* could in principle be recruited to change parameters or structure of the HA algorithm. With a sample rate of 16 kHz and a 16-bit code per sample, about one million bits of acoustic data get recorded every four seconds by the hearing aid. One could imagine that machine-learning methods take advantage of these in situ acquired acoustic data streams, e.g., to train an environmental classifier.

In general, the field of *machine learning* refers to methods that aim to improve the

*Corresponding author: bdevries@gnresound.com

Proceedings of ISAAR 2013: Auditory Plasticity – Listening with the Brain. 4th symposium on Auditory and Audiological Research. August 2013, Nyborg, Denmark. Edited by T. Dau, S. Santurette, J. C. Dalsgaard, L. Tranebjærg, T. Andersen, and T. Poulsen. ISBN: 978-87-990013-4-7. The Danavox Jubilee Foundation, 2014.

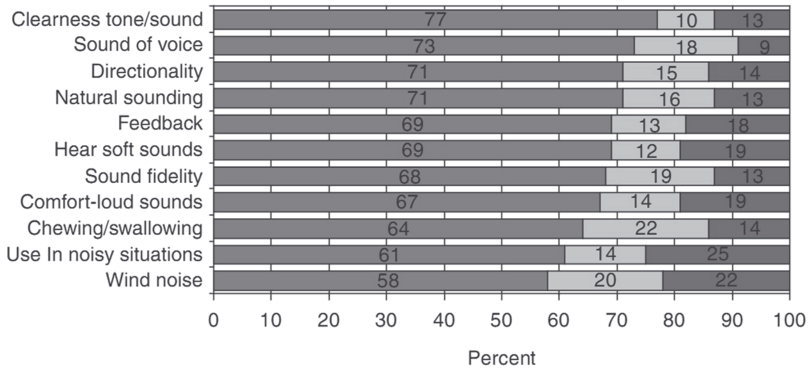

Fig. 1: Percentages of patient satisfaction with sound processing in hearing aids. Figure from Kochkin *et al.* (2010).

performance of a device through (learning from) experience with that device. In the hearing-aid context, in situ updates of the algorithm by patient preference feedback or by the acoustic environment could be considered machine-learning-based design. Today, HA design is almost exclusively the domain of the professionals. Why has in situ machine-learning-based design not yet claimed a substantial role in the HA design process? In this paper we will discuss some fundamental signal-processing issues that hinder application of machine-learning-based design to HA algorithms.

WHY IN SITU MACHINE LEARNING?

How satisfied are users of hearing aids? The graph in Fig. 1 is from a large study in 2010 on the hearing-aids market by Kochkin *et al.* (2010). The horizontal bars represent patient satisfaction rates with various aspects of sound processing in hearing aids. The bars on the left reflects the percentage of people that are happy, dark grey (right) indicates dissatisfaction, and light grey (middle) relates to a neutral opinion. Let's keep this simple: about 20% of hearing-aid patients are not happy with the sound processing performance of their devices.

This is a remarkable number because over the past decade, we, the engineers and scientists in the hearing-aids industry and in academic environments, have collectively spent a few thousand man-years on improving the sound processing in hearing aids. Apparently, despite a very extensive collective engineering effort, one out of five patients remains not satisfied. The performance of sound processing in hearing aids seems to have plateaued.

There is a plausible explanation for this observation. When an engineer designs a hearing aid, he does not know yet who the patient will be, he doesn't know the hearing-loss portrait of that patient, nor does he know in which acoustic environments the patient will spend his time. To complicate matters, this type of knowledge changes

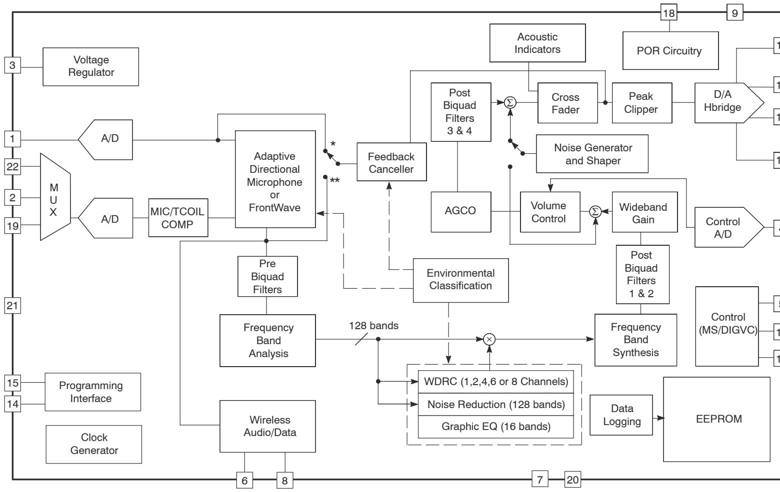

Fig. 2: Block diagram of the AYRE-SA3291 hearing-aid algorithm, ON-Semiconductor (2013).

over time. Every time a patient puts in his hearing aid, the physical placement of the device will be a bit different from last time, leading to an altered acoustical situation in and around the ear. In other words, when the engineer designs the sound-processing properties of a hearing aid, he has to deal with many unknowns about the actual circumstances where the hearing aid will be used. It won’t help to ask the engineer to work harder or do his extra very best this time, since these future in situ conditions are simply unknown. Instead, we must provide the patient will tools to solve problems right there when they occur on the spot.

In order to get an idea of what we are up against, have a look at Fig. 2, which is a block diagram of a commercial hearing-aid algorithm by ON-Semiconductor (2013). We use this particular block diagram because it is publicly available but the discussion applies generally to the signal-processing algorithms of commercially available hearing aids. Most blocks in this graph hide sub-algorithms that are at least as complex as this top-level diagram. Now suppose that you are at a cocktail party and you can’t understand your conversation partner. You would like to make a small change to this circuit and test a few variants, but how? If you pull a wire, this circuit will likely crash and no output get generated. Which wire should you pull anyway? Or should you add a wire somewhere? In order words, how do you bring about variation as a means for experimentation in this system? Even if you succeed in improving the quality for your current situation, will that change still be an improvement later, after the party is over? In practice, the way to update systems like this one is to give it to an expert signal-processing engineer and let him tinker with it; then take it back after a few months and hope that it works better. But that's not what we are interested in here. If this system doesn't work to your full satisfaction in the field, you might be willing to invest maximally one minute to make it sound better. And it should sound better because if it doesn't, you will be less willing to spend that minute the next time. This signal processing circuit is not fit for that purpose.

Based on the foregoing discussion, let us state an important challenge for the HA industry. How can we build tools that facilitate *fast and easy (machine-learning-based) re-design of hearing-aid algorithms driven by end users and the environment*. There are three very challenging aspects about this goal. Whereas a normal design update by a signal-processing expert in his laboratory environment may take a few months, in this challenge the aim is to execute an incremental design update (1) by a (non-expert) user, (2) under normal operational conditions, and (3) within a minute. In search for answers, we are inspired by research from others on how the brain processes information. In this paper we will discuss some aspects of computation in brains that in our opinion should influence the future engineering practice of HA algorithm design. However, before we turn to the brain, let us discuss an important engineering lesson for the design of systems with large uncertainties.

DESIGN FOR REDESIGN – THE FLIGHT OF THE GOSSAMER CONDOR

In 1959, the British industrialist Henry Kremer announced a prize of £$50,000$ (in today's money worth about 1 million euros) for the first successful human-powered flight around a figure-eight course with the two turning points placed half a mile apart. A second prize of £$100,000$ was created for the first human-powered flight across the English channel.

Many years and 50 failed attempts passed. In 1977, the British aviation engineer Paul MacCready took on the challenge and noticed a common pattern when he studied the records of past attempts. Previous engineering teams had often invested more than a

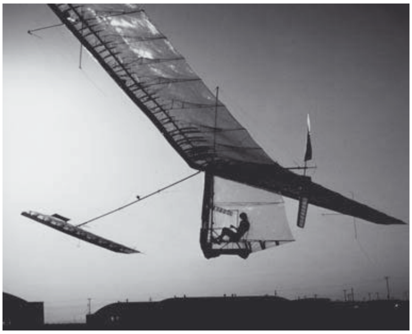

Fig. 3: The Gossamer Albatross, the first human-powered airplane that crossed the English channel. Figure from Raskin (2011).

year to carefully design a prototype plane based on elaborate theories and conjecture. Then, a few seconds after take-off of the maiden flight, a year's work would crash on the ground and obliterate the massive effort.

MacCready came to a crucial insight. The past efforts were focused on solving the wrong problem. The essential problem was not how to design a human-powered airplane. Instead, the essential problem was that *they did not understand the problem* (Raskin, 2011). Rather than attempting to design an optimal aircraft, MacCready reformulated the problem as the quest to design an airplane that could be re-built in hours, not months. His team started building planes from cheap and light aluminum tubing, mylar, wires, and scotch tape. In MacCready's approach, design was to be interpreted as an experiment to learn more about the problem. The first flight failed right away. But the team learned from the crash and delivered a second prototype just a few hours later. This process of fast iterative redesign continued for about half a year until 23 August 1977, when Bryan Allen of MacCready's team pedaled the Gossamer Condor for the 223rd time and cleared the finish line 7 minutes and 27 seconds after take-off. Two years later, Allen flew a further evolution of the Gossamer (the 'Albatross') across the English channel to claim the second Kremer prize, cf. Fig. 3.

Where other teams had failed for more than 17 years, MacCready's fast-iterations approach turned out to be the key to solving poorly-understood engineering problems. While this story has on the surface little to do with hearing-aid design, the underlying challenge to cope with a poorly-understood problem is the same for both tasks. This story illuminates the *engineering need* to focus on fast redesign of hearing-aid sound-processing algorithms, instead of a research focus on the optimal algorithm per se.

INFORMATION PROCESSING AND THE BRAIN

Engineers study the brain for its usability to design artificial systems. Since the brain is our most crucial instrument in our drive to survive, it must work today and yet be fully prepared to adapt to unforeseen new circumstances. In the next sections we will discuss a few salient properties that enable the brain to execute fast redesign iterations so as to cope with a world where the problems keep changing in unpredictable ways.

Probability theory

If the brain is a system that processes information then there must be some computing rules that the brain adheres to. There is strong scientific support for the claim that brains compute with the rules of probability theory (e.g., Friston, 2009). This is the same probability theory that we all got to love and hate in high school.

We can use probability theory to predict the future, based on observations from the past. For instance, if we observe 100 coin tosses and 96 out of 100 throws came up tails, then we predict that the 101st observation will come up tails with higher probability than for heads. Intuitively this happens by extrapolating past observations.

Technically, in order to predict the future we need to build a *model* to summarize regularities that were present in past observations and use that model to predict the future. We humans need to have some capacity to predict the future, because we want to avoid to be surprised by the physical world around us. For instance, we must be able to make predictions on what's edible or hostile to us. More generally, any large surprise in the physical world could possibly kill us. So, a key task of the brain is to build a model for the world in which we live and use that model to make predictions about that world.

Probability theory can be used to make optimal predictions about future (data) observations by

$$\underbrace{\Pr(\text{future}\,|\,\text{data})}_{\text{data-based prediction of future}} = \sum_{\text{all models}} \underbrace{\Pr(\text{future}\,|\,\text{model})}_{\text{model-based prediction of future}} \times \underbrace{\Pr(\text{model}\,|\,\text{data})}_{\text{model based on past observations}} \qquad \text{(Eq. 1)}$$

The expression Pr(.) here is mathematical notation for a probability mass function, but we will not bother with explaining the details of the formula, other than to point out that something as complex as predicting the future can be captured by a single-line equation. The left-hand side states that we want to predict the future from past data. The data refer to observations from the outside world that enter the brain through sensory organs like the eyes or ears. The right-hand side states how predictions of data relate to a model and past observations. The model can be implemented by a brain or by a computer program. The right-most factor, $\Pr(\text{model}\,|\,\text{data})$, captures what the model has learned from past data. By another rather simple manipulation with probability theory we can express how models learn from data:

$$\underbrace{\Pr(\text{model}\,|\,\text{data})}_{\text{model after learning}} = \frac{\overbrace{\Pr(\text{data}\,|\,\text{model})}^{\text{model based predictions}} \times \overbrace{\Pr(\text{model})}^{\text{model before learning}}}{\underbrace{\Pr(\text{data})}_{\text{evidence}}} \qquad \text{(Eq. 2)}$$

In probability theory this equation is known as *Bayes rule*. Bayes rule describes how we learn about the world. It doesn't matter if the observations relate to music, video, or even financial stock rates: Bayes rule applies and tells us how to optimally update our knowledge about a phenomenon based on new observations about that phenomenon. Bayes rule is basically a prediction-correction method. The model gets updated on the basis of differences between actual and (synthesized) predicted observations.

If a human brain were capable of executing Bayes rule, then our concept of what a tree looks like would get updated every time when we see a tree. The more trees we see, the better we understand what a tree looks like. It seems that it would be very useful for a brain to be able to process sensory information by Bayes rule, because it would enable us to learn a model about the world just by looking at the world. Apparently, using the same rules from probability theory we can then use that model to make predictions about the world, which are so crucial for us to stay alive.

It can be shown that, under some very agreeable assumptions, Bayes rule prescribes the *optimal* method for learning from observations (Jaynes, 2003). So there is no need to look for a specialized learning algorithm that works particularly well for any specific problem. The simplicity of Bayes rule is a strength. Whether we have to learn a language or learn about how to repair a bicycle, Bayes rule is how we *should* learn. If the brain computes with probability theory then there is no need to invent new prediction or detection methods when the outside world changes. It doesn't matter if the observed signals are of acoustic or visual nature, the difficulty lies mostly in how to *implement* Bayes rule, both in brains and computers.

The probabilities that we discussed relate data to models and back. Models and data are very much the core issues for engineered signal-processing systems. Next we take a look at how the brain deals with models from the perspective of adaptability.

Models and structures

Signal-processing algorithms can be intuitively visualized by block diagrams like in Fig. 2. A block diagram consists of a set of blocks (nodes) and links (edges) that connect the blocks. With each link we associate a variable in the system. In a block, mathematical relations between the connected variables are described. Often, we may find another block diagram in a block, so blocks can be used to hide details of the algorithm. The algorithm *structure* refers to the mathematical relations between the variables that are described by a block diagram. We also like to distinguish between variables whose values change as time moves on (the *state* variables) and those (the *parameters*) whose values are expected to stay fixed or change much slower than the rate of change of the states. In neural terms, the structure relates to the neuronal network of the brain, the parameters are represented by the strength of synaptic connections between neurons, and the state relates to the electric fields in the brain. In particular, our perception of the world is represented by the state variables. The model structure and parameter values provide constraints on how the states (read: our perception) will change over time. If our perceptions and prediction of future perceptions are accurate enough, we can stay alive.

Unfortunately, unexpected things will happen and we will need to change the algorithm structure and parameter values so as to keep our model of the world sufficiently accurate.

It is clear that if we change a structure at one location, we do not want that change to have serious consequences on variables in another location of the network. If the network were now to be adapted at the latter location, this could have effects elsewhere again and thus lead to a *snowball* effect of unpredictable changes, likely followed by a crash of the algorithm. Therefore, *modularity* is an essential characteristic of complex yet adaptable networks. A modular network is composed of sub-networks called modules with more dependencies within the modules than between the modules. The relative independence of modules prevents the snowball effect of changes to escalate.

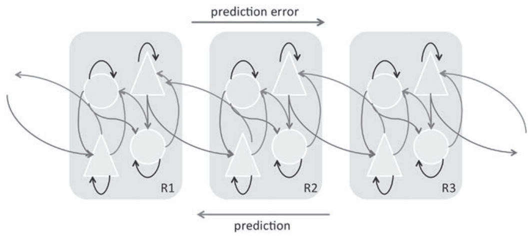

Fig. 4: An example flow graph of hierarchical modularity across three cortical regions. Figure from Friston (2009).

On the other hand, some communication between modules is necessary to generate behavior that transcends the functional complexity of individual modules. In order to avoid the snowball effect, modules should preferably depend on other modules that are *more stable* than themselves. Let's assume the opposite, namely that module A depends on module B and the natural rate of change for B is faster than for A. In that case, A will have to adapt each time that B changes, which is more often than A's natural rate of change. The idea that the snowball-of-changes effect can be avoided by constraining intermodule communication to flow from more to less stable structures leads to *hierarchical* networks.

Technically, probability theory supports hierarchical modularity almost effortlessly. Bayes rule decomposes into a hierarchy of four modules by

$$
\begin{aligned}
&\Pr(\text{ model}\,|\,\text{data}) \propto \Pr(\text{data}\,|\text{ model}) \times \Pr(\text{ model}) = && \text{(Eq. 3)}\\
&\quad \Pr(\text{data}\,|\,\text{states}\,,\text{parameters}\,,\text{structure}) && \text{(now)}\\
&\times \Pr(\text{states}\,|\,\text{parameters}\,,\text{structure}) && \text{(short-term memory)}\\
&\times \Pr(\text{parameters}\,|\,\text{structure}) && \text{(mid-term)}\\
&\times \Pr(\text{structure}) && \text{(long-term)}
\end{aligned}
$$

In the final result of the computation, the left-hand side $\Pr(\text{ model}\,|\,\text{data})$, the model depends directly on fast fluctuations in the observed data. Straight implementation leads to an undesired network structure. However, after the hierarchical decomposition, at each level, variables only depend on other variables that are more stable than themselves. We now have an answer to our question on *how* to implement Bayes rule. Through hierarchical modularity the snowball effect of changes is avoided. This property is crucial when in situ structural algorithm changes are demanded.

We can think of many reasons why modularity is the most prominent feature of adaptable systems. But how would the brain know that? Is there an evolutionary drive for brains to develop modular structures? If we accept that the brain is mostly an engine for probabilistic reasoning, then it would help if probability theory would prefer modular over densely coupled structures (all else being equal). This is indeed the case. The factor $\Pr(\,\mathrm{data}\,)$ in Bayes rule, known as the *evidence*, can be used to evaluate how well a model summarizes a set of observations. It can be mathematically shown that the (logarithm of the) evidence decomposes into a sum of two terms, namely *accuracy* plus *model simplicity*:

$$\begin{aligned} \log(\,\mathrm{evidence}\,) &= \mathrm{accuracy} + \mathrm{simplicity} \\ &\approx \text{‘works today’} + \text{‘works tomorrow’} \end{aligned} \qquad \text{(Eq. 4)}$$

The first term, accuracy, measures how well the model predicts past observations. If we need to predict future observations, it makes sense that we prefer models that performed best on past data, so we want models with high accuracy. A system that scores high at accuracy works well today. However, the second term, the simplicity term, favors models that are simple and adaptable. Indeed, it can be shown that modular structures score higher simplicity values than densely coupled systems. As we discussed, modular systems are more adaptable than coupled systems. Therefore, probability theory prefers structures that balance excellent performance today (high accuracy) against adaptability for tomorrow (high simplicity). We also conclude that if brains would follow probability theory, then it's no surprise that brains are both excellent performers today and yet remain very adaptable. After all, both properties are highly prioritized by straight probability theory. The brain has no choice but to optimize both for today *and* an unknown future.

Driven by data

We discussed how probabilities and models relate to information processing in the brain. The third term in the equations for learning and prediction is called the data or observations.

Data are observed through sight, hearing, taste, smell, and touch, collectively known as our senses. Observations inform us about the current state of the world around us. We use probability theory to summarize observations in models and use these models to predict how the world evolves.

One of the most interesting aspects of our brain is how much we seem to learn from just a few teaching events. After a mother has showed her two-year old daughter a few times what a tree looks like, the girl is able to identify new trees that she has not seen before and also to discriminate trees from other plants in general. Considering the various shapes, sizes, and colors that apply to trees, it would be impossible for a child to learn to reliably recognize trees from just a few remarks by her mother. Instead, a child learns what trees look like through building models straight from incoming visual data. There is no teacher involved here. The interaction with her mother just added a label (‘tree’) to the concept of a tree that had already been acquired through modeling the world in an unconscious fashion. In the machine-learning field, learning without a teacher is called ‘unsupervised learning’.

The human cortex holds about 10^{14} configurable synapses, which can be considered

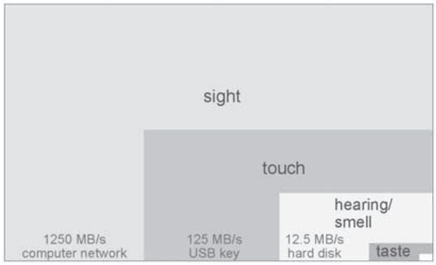

Fig. 5: Data rates for the senses relative to bandwidth of computer networks (McCandles, 2010).

parameters of the brain. We live about 10^9 seconds, so on average there is room to train about $100,000$ synapses every second. Indeed, brains receive a massive amount of data through the senses, for instance the retina sends more than 10 million bits of data to the brain every second. The role of teachers, parents, books, and other sources of abstract information is mostly to help us sort out which parts of these incoming data streams are important or should be ignored. In other words, teachers help us to select and label data streams that are used to train a model of the world. Crucially, in order to cope with a world where the settings and problems keep changing, a massive amount of unsupervised learning must always be going on.

In Fig. 5, data visualization artist Dave McCandles, based on work by the Danish science writer Tor Nørretranders, graphically displayed the amount of information that the different senses pass on to the brain in comparison to the bandwidth of computer networks (McCandles, 2010). Clearly, vision is the dominant sense. The white box in the lower-right corner represents the amount of data (0.7%) that is processed consciously in relation to the colored planes that refer to unconscious processing. Apparently, almost all incoming data is processed unconsciously. Building models of the world, including creating a model for what a tree looks like, is mostly an unconscious process.

Summary of information processing in the brain

In order to adapt to unforeseen changing conditions, brains need to iterate quickly through new model proposals for explaining the world. In the past three sections on information processing in the brain, we found three crucial ingredients for iterating quickly though signal processing system proposals. The first ingredient concerns probability theory as a foundational calculus. Probability theory prescribes how to learn and predict in a world where noise obscures the signals, where observations are scarce and where people's preferences change. The second principle relates to hierarchical modularity. In order to discover better algorithms, we need to test alternatives to existing algorithms and at the same time remain operational. We can only introduce a change to an existing algorithm if the effect of the change does not cause other parts of the algorithm to crash. We must survive the change and modularity is a crucial structural element so as to limit the impact of changes throughout the algorithm. Finally, when talking about the data we noted that the structure of real world data is so rich and volatile that we cannot rely on teachers, parents, scientists, and engineers to design and update the algorithm. Surviving in the real world implies a massive amount of unsupervised learning, which is always going on in the background. In engineering terms, continuous calibration is essential.

HEARING-AID SYSTEMS THAT WORK TODAY AND TOMORROW

Most of this paper has been dedicated to a review of data processing in the brain. Let us now get back to the engineering practice. We left this topic about half an hour ago when we were stuck with a block diagram of a hearing-aid algorithm. You were at a party and did not understand your conversation partner. You then wanted to test some variants of the hearing- aid algorithm right there when the problem occurred, but the system looked so complicated that any ideas on how to change the circuit were hard to come by. Our feeling was that if you would change anything, the algorithm would probably crash.

But let us assume that you have managed the dependencies between modules in such a way that you have enough confidence that a small change will not kill the algorithm. Then you can introduce some small changes to the hearing-aid algorithm and with a bit of luck you can improve your listening experience at the party. The next question is now whether the hearing aid should stick to this new configuration after you have left the party. You gave the hearing aid some new information, namely you showed the hearing aid how to behave when you are at a cocktail party. How relevant was that information for other acoustic environments? When the party is over, and you are in your car driving home, the hearing aid has two possible algorithms to choose from: the one that you came to the party with and the other algorithm that you preferred while the party was alive. Since you don't want to keep fiddling with your hearing aid every time when something changes in the acoustic environment, we want the hearing aid to decide for you.

In order to answer this question, the hearing aid would have to consider what features of the cocktail-party environment were so favorable for the second rather than the first algorithm and it would have to consider if or how much of these features remain active in the current car environment. In other words, the hearing aid should have access to a model of the acoustic world and it should be capable to answer what-if questions based on information that is preserved by the model. The hearing aid should have built such a world model by unsupervised training on past acoustic observations. In principle, this seems possible since a hearing-aid microphone records one million bits of acoustic data every four seconds. This continuous data stream should be summarized by a hierarchically organized structure, which is a necessary ingredient for the model to stay changeable, so it *can* adapt as new data get recorded. We have also discussed that the model should practice Bayesian reasoning in order to assess *how much* to adapt.

Unfortunately, these necessary ingredients for in situ learning are not part of today's HA signal-processing algorithms. As a result, rational adaptation of HA algorithms based on in situ acquired evidence is limited today. In our opinion, the key hearing-aid signal-processing challenge for the next decade will be to absorb the discussed additional features into our algorithms.

DISCUSSION

In this paper, we have taken a high-level perspective on the design and in situ re-design of hearing-aid algorithms. We have tried to make an argument for why fast in situ re-design of HA algorithms is crucial if we want to break through the 20% barrier of unsatisfied end users. Wireless links to remote control devices and fancy user interfaces lead to impressive products, but in the end all patient interactions should result in rational algorithm updates based on the evidence. As it turns out, while today's hearing-aid algorithms keep roughly 80% of end users satisfied, they are not suited for fast in situ experimentation and adaptation in case the patient is not happy. We then identified three salient properties of a very successful adaptable and personalized signal-processing system, namely the brain, that are absent in today's HA signal-processing structures. Specifically we discussed (1) learning through strict application of probability theory, (2) a hierarchically modular algorithm structure, and (3) continuous calibration. The absence of these properties hinder machine-learning-based re-design of today's HA algorithms. On the other hand, an emerging trend of cross-fertilization of ideas between the computational neuroscience, machine-learning, and signal-processing communities should make us mildly optimistic that significant progress towards in situ HA design can be achieved over the next decade.

REFERENCES

Friston, K. (**2009**). "The free-energy principle: a rough guide to the brain?" Trends Cogn. Sci., **13**, 293-301.

Jaynes, E.T. (**2003**). *Probability Theory: The Logic of Science*. Cambridge University Press.

Kochkin, S. (**2010**). "MarkeTrak VIII: Consumer satisfaction with hearing aids is slowly increasing," Hearing Journal, **63**, 19-20,22,24,26,28,30-32.

McCandles, D. (**2010**). "The beauty of data visualization", Talk at TED Global Oxford (`http://goo.gl/7MzQ`). Based on work by Tor Nørretranders.

ON Semiconductor (**2013**). "Datasheet for AYRE SA3291 Preconfigured DSP System for Hearing Aids," `http://onsemi.com`.

Raskin, A. (**2011**). *Wanna Solve Impossible Problems? Find Ways to Fail Quicker*. `http://goo.gl/9zX3L`.