Sonification of Probabilistic Feedback through Granular Synthesis

John Williamson
University of Glasgow

Roderick Murray-Smith
University of Glasgow
Hamilton Institute

We describe a method to improve user feedback, specifically the display of time-varying probabilistic information, through asynchronous granular synthesis. We have applied these techniques to challenging control problems as well as to the sonification of online probabilistic gesture recognition. We're using these displays in mobile, gestural interfaces where visual display is often impractical.

Human interactions with systems are difficult for two reasons: Users misinterpret systems, and systems misinterpret users. Solutions to the latter problem are the realm of recognition technologies and inference algorithms, but we suggest that such sophisticated interpretive mechanisms will be most fruitful if they're combined with technologies to improve human understanding of the interaction. In essence, we're interested in improving the user feedback that systems provide—especially when visual displays might be impractical—particularly when we can enhance the interaction experience by displaying the interaction's changing state, accurately displaying uncertainty and incorporating predictive power.

A human–computer interface interprets user actions and carries out the user's intention. In practice, a system can't interpret a user's intention with absolute certainty; all systems have some level of ambiguity. Conventionally, system designers ignore this; however, by explicitly representing the ambiguity and feeding it back to the user, we can increase the interaction quality.

Ambiguity in interfaces, which Mankoff et al. have described,[1,2] is a particularly significant issue when the system's interpretations are complex and hidden from the user. Users can intuitively manipulate a system's physical buttons because they have no hidden complexity. Our goal is to design systems having all the power of the most advanced recognition algorithms but which remain as intuitive as a simple mechanical button. Explicit uncertainty in the interface is a step toward this goal by making the true state of the inner processes of the system visible to the user.

In this article, we describe asynchronous granular synthesis, a method for displaying time-varying probabilistic information to users. Granular synthesis is a technique that mixes together short segments of sounds. The segments are drawn according to a probability distribution, which gives an elegant link to probabilistic models in the user interface. We extend the basic synthesis technique to include distribution over waveform source, spatial position, pitch, and time inside waveforms. To enhance the synthesis in interactive contexts, we "quicken" the display by integrating predictions of future user behavior.

These techniques can be used to improve user performance in continuous control systems. As an example, we explain the feedback technique that we've implemented with asynchronous granular synthesis on a prototype system for helicopter flight simulation, as well as for a number of other example domains.

Ambiguous interfaces

Representing ambiguity is especially significant in closed-loop continuous-control situations, where the user is constantly interacting with the system to achieve some goal (see Figure 1, next page).[3] Formulating the ambiguity in a probabilistic framework, we consider the conditional probability density functions of sequences of actions associated with each potential goal in the system. Examples of goals, such as those in a workstation interaction task, might be actions such as Open File or Exit. In this case, ambiguity might arise when the system must interpret a mouse click close to a border between menu items.

In our model, users try to maximize the system's "belief" about the goal they want to

1070-986X/05/$20.00 © 2005 IEEE

Published by the IEEE Computer Society

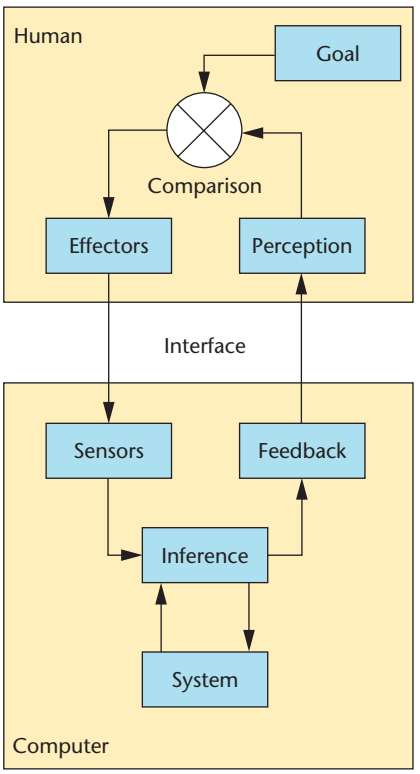

Figure 1. Model of user interaction in a closed-loop system. We concentrate on the right-hand side of the diagram, where feedback from the inference process is provided to users and which they can then compare with their goals.

achieve, by interacting with the system via the various input modalities available to them. For example, in a conventional desktop interface, users might move the mouse toward an icon they wish to select. The system uses its model of user behavior to update its beliefs and feeds back the inference so that users can act to correct any potential misinterpretation. This feedback represents the complete current state of the system's interpretation of the users' inputs—for example, the current likelihood of each potential gesture in a gesture recognition system. When the probability of an action is sufficiently high, the system can act appropriately. In this context, giving feedback about the probabilities of each potential action or goal can help users, especially if the system's display has predictive power and can display the sensitivity of future states to current actions to answer the question, What must a user do to increase the probability of achieving an intended goal? To help users select the most appropriate system action to achieve their goal, we propose to sonify (to make audible) the time-varying properties of goal state distribution.

Audio feedback

Audio is suitable for presenting users with high-dimensional, dynamic data, especially when the users' eyes might be occupied. This would be the case, for example, with multiple visual displays such as those a helicopter pilot would be concerned with to maintain the aircraft's stability, or with simpler tasks such as walking while using a mobile device (such as a PDA).

A continuous-feedback model requires that the model dynamically update the probability of each goal—such as potential gestures—in real time, and transform it to an audio representation of the changing probabilities. At each discrete time step t, a vector of conditional probabilities $P(\text{goal} \mid \text{input}) = P(G \mid I) = [p_1, p_2, \ldots p_n]$ is updated and displayed in audio. Any suitable inference mechanism can perform the update, so long as the model can evaluate the probability distribution in real time.

A particularly suitable method for translating the probability distribution to an audible format is sonification via asynchronous granular synthesis.

Granular synthesis

Granular synthesis is a probabilistic sound-generation method, based on drawing (extracting) short (10–500 ms) packets of sound, called grains or granules, from source waveforms. See the literature for more information,[4-7] especially the comprehensive overview by Curtis Roads.[8] A large number of such packets are continuously drawn from n sources, where n is the number of elements in the probability vector.

For the discrete case (for example, in a user interface where we are selecting items from a list), we can either synthesize or pre-record these waveforms. In the case of a continuous probability distribution, where n is infinite, the sources must have a continuous parametric form, which we generate in real time as the grains are drawn. For example, we could use frequency modulation (FM) synthesis to represent a one-dimensional continuous distribution with the modulation index as the source parameter. After we draw the grains, we envelope them with a smooth window to avoid discontinuities and sum them into an output stream. Figure 2 shows the basic process.

In asynchronous granular synthesis, the grains are drawn according to a distribution giving the probability of the next grain's being selected from one of the potential sources. This gives a discrete approximation to the true distribution. Because the process is asynchronous, the grains' relative timing is uncorrelated.

The grain durations we used in our various implementations are of the order of 80–300 ms

with squared exponential envelopes; this produces a smooth, textured sound at higher grain densities. Our engine system generates grains such that between 100 and 1,000 are always active, for a relatively accurate representation; as one grain finishes, a new one is drawn with some probability. This implies an exponential distribution on the time to generate a new grain, and it's from this process that the asynchronicity of the synthesis arises.

Asynchronous granular synthesis gives a smooth continuous texture, the properties of which we modify by changing the probabilities associated with each grain source. Even in situations where other synthesis techniques could be used, granular synthesis gives strong, pleasing textures, formed from the enormous number of audio particles, which are easily manipulated by a system designer in an elegant and intuitive manner.

It also has the advantage that a distribution can be defined over time inside the source waveform, defining the probability of a grain being drawn from a specific time index in the wave. This allows for effective probabilistic time-stretching, which is a powerful tool in system–user interactions where it's important that a user make progress toward achieving some goal (for example, recognition of a relatively complex gesture, such as for a compound action in a user interface). Similar distributions over pitch (playback rate) and spatial position can also be defined.

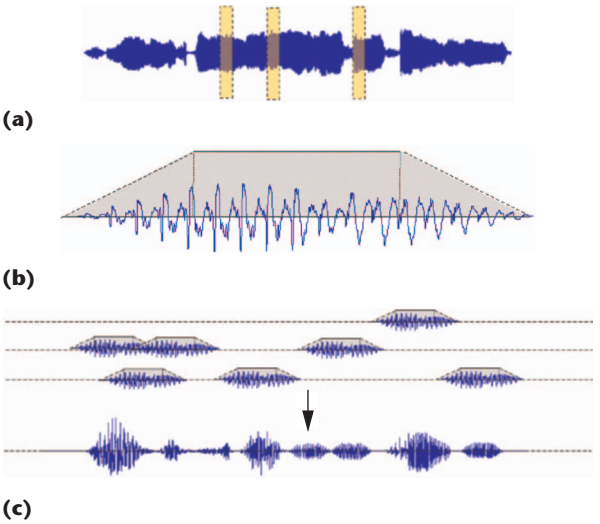

Figure 2. Simple granular synthesis process. (a) Grains are drawn from the source waveforms, according to their probability distributions. (b) Each grain selected in (a) has an amplitude envelope applied. (c) All the grains that are currently active are summed together into a final output waveform.

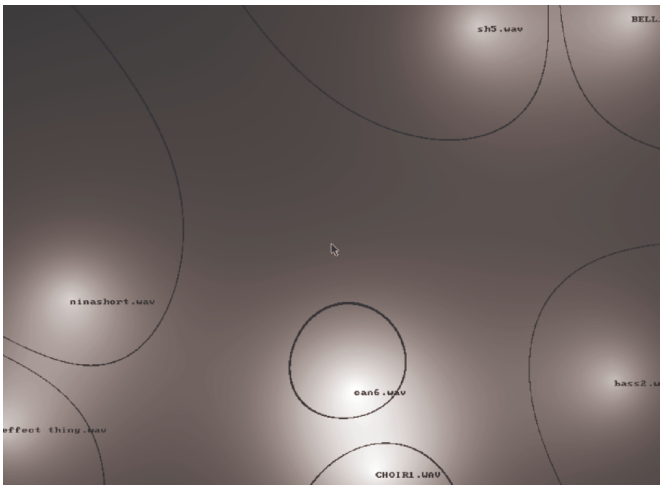

Figure 3. Mixture of Gaussian densities in a 2D state space illustrates the basic concept. Each Gaussian is associated with a waveform, which is shown at its center. The user moves the mouse pointer in this window to control the sound output.

State space representation

Figure 3 shows how we could use a mixture of Gaussian densities to map regions of a 2D state space to sound. Each Gaussian density is associated with a specific sound, and as the user navigates the on-screen space, the timbre of the sound changes appropriately, with the texture associated with each goal becoming more prominent as the cursor approaches. Although here the densities are in a simple spatial configuration, we can apply the technique to any state space representation by placing appropriate densities in the space. In a dynamic system, such as a helicopter flight simulator (which we discuss later), we could place the densities on significant regions such as equilibrium points or along transients.

Gesture recognition

We can extend the sonification technique to gesture recognition as an example without explicit state space representation. We can sonify a probabilistic gesture recognizer by associating each gesture model (in our helicopter flight simulator implementation, hidden Markov models) with a source waveform, and each model's output probability then directly maps to the probability of drawing a grain from the source corresponding to that model (see Figure 4, next page). The temporal distribution of grains inside

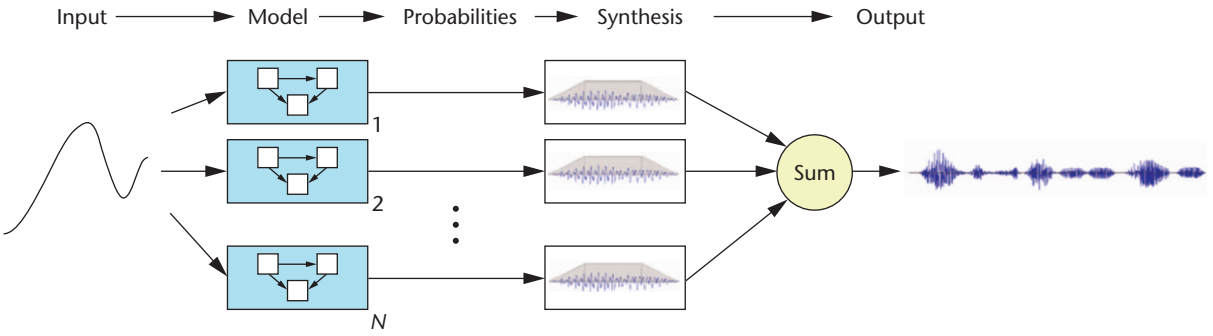

Figure 4. Mapping from an input trajectory to an audio display via a number of gesture recognition models. Each gesture is associated with a model, and the output probabilities are fed to the synthesis algorithm.

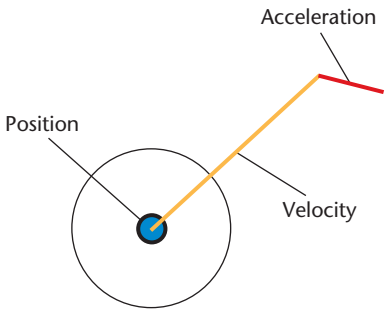

Figure 5. Visual example of quickening. Here, the controlled object is the inner circle. The estimated velocity and acceleration vectors are shown. Similar visual displays have been suggested for use in aiding helicopter pilots in maneuvering.[11]

the source waveforms maps to the system's estimate of user progress through the gesture.

The design issue is then reduced to creating a suitable probability model and selecting appropriate waveforms as sources. The overall grain density remains constant throughout the sonification. In practice, this produces a sound that's incoherent when ambiguity is high, resolving to a clear, distinct sound as system recognition progresses. The sonification's primary effect is to display the current goal distribution's entropy.

Display quickening

Quickening[9,10] is the general process of adding predictions of future states to a display. In manual control problems, this lets users improve their performance; it's been shown that the addition of even basic predictive power can significantly improve the performance in difficult control problems. For example, Charles Kelley describes quickening submarine controls to show future changes to the orientation of the ship that control actions will have. He notes:

> Experience indicates that, by using a properly designed predictor instrument, a novice can, in ten minutes or less, learn to operate a complex and difficult control system as well as or better than even the most highly skilled operator using standard indicators.[9]

Such techniques are directly applicable to real-time sonifications of probabilistic state in interactive systems. Giving the user information on the sensitivity of goals to inputs can allow faster and more robust exploration and control to be achieved.

For the purposes of our framework, we want to evaluate $P(G_{t+T} | I_1 \ldots t)$ where T is a time horizon, and t is the current time. This full distribution is usually computationally intractable, so we make simplifying approximations.

The most basic quickening technique is to display the derivative of the variables under control (see Figure 5). In our framework, the variables are the time-varying probabilities. Displaying the gradient of the density along with its current value can improve the feedback's effectivity as users perceive whether their actions are moving them toward, or away from, hypothesized goals in a continuous manner. The prediction takes place directly in the inferred space, and so we can easily apply it to any problem. However, this does assume that linear predictions are meaningful in the goal space.

We can quicken the granular audio display by taking the first, second, and higher derivatives of each probability p with respect to time and then forming the sum

$$v = p + \sum_{i=1}^{n} k_i \frac{dp^i}{d^i t}$$

where i is the order of the derivative and k_i are scaling factors. We saturate v to clip it to the range (0, 1). We can then treat this value as a probabil-

ity and directly sonify it using the granular synthesis process we've described. When the user increases the probability of achieving a goal, the proportion of grains drawn from the source associated with this goal likewise increases; similarly, the proportion is decreased as the goal becomes less likely. In practice, higher-order derivatives are generally less intuitive from the user's point of view as they relate very indirectly to the variable under control, and are also likely to be dominated by noise unless special care is taken in filtering the signals.

As a simple practical example of a quickened display, we augmented the display shown in Figure 3 to include first-order linear predictions. This aids users as they explore the goal space by rapidly increasing the intensity of the system's feedback. As the users move toward the center of a goal, they can quickly determine which direction will give the greatest increase in probability. In particular, the quickening helps users accurately ascertain the distribution's modes.

As a user approaches and then overshoots the mode in the mixture-of-Gaussians example, there is a rapid increase followed by an equally rapid decrease in the feedback intensity for that goal, allowing for faster and more accurate targeting of the modes. In higher-dimensional exploration tasks, such as examining multivariate data sets, the quickening is particularly useful for finding subtle gradients that might be difficult to perceive with an unaugmented display. As the dimension increases, increasing the weighting k_i of the derivatives can help compensate for the spreading out of the density.

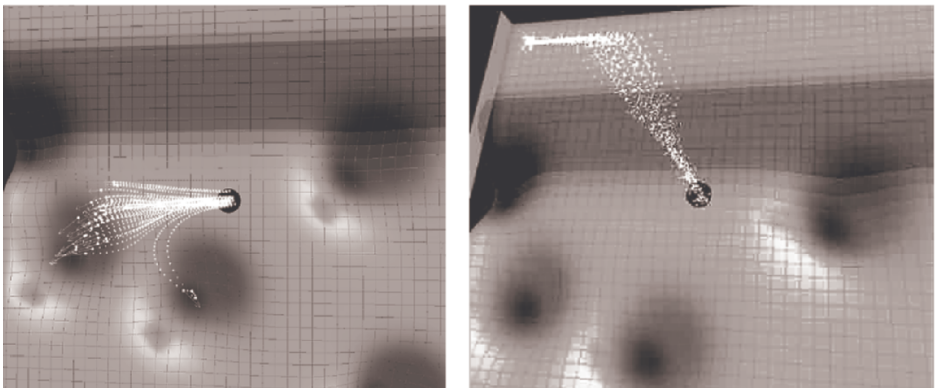

Figure 6. Monte Carlo sampling of the distribution of future states in a simple control task. In this case, the future states represent the potential positions of a ball bearing rolling across a nonlinear landscape. Each sample at the horizon (the arrows, which appear as little white lines) corresponds to an output grain. Each depression has a sound source associated with its mode. The darker region has higher model uncertainty. The right-hand image shows where potential future trajectories of the ball bearing will pass through a region of high uncertainty.

Monte Carlo sampling for time-series prediction

Monte Carlo sampling is a common statistical method for approximating probability densities by drawing a number of discrete samples from the probability density function. This often leads to more tractable computations than directly working with the target distribution. For example, we can use it to approximate $F(p(x))$, where $p(x)$ is a (potentially complex) distribution, and F is a nonlinear function.

There's a particularly elegant link between granular synthesis and Monte Carlo sampling of probability distributions—we can directly map each sample taken in the process to a single grain in the output. Few other sonification methods would be suitable for directly representing the output of the Monte Carlo sampling process; approximations to the distribution would be required. Where there are more grains than samples, we can match grains to samples in a round-robin manner. Thomas Hermann et al.[12] describe a particulate approach to sonification of Markov Chain Monte Carlo simulations to display the properties of the sampling process. For our purposes, we concentrate on using Monte Carlo sampling for time-series prediction.

For example, given a model of the dynamics of a particular interactive system, where there may be both uncertainty in the current state and uncertainty in the model, Monte Carlo sampling can approximate the distribution of states at some point in the future. We can do this by drawing a number of samples around the current state, and propagating these forward according to a model of the system dynamics, with appropriate noise at each step to account for the uncertainty in the model.

Simple dynamic system

As a practical example of the Monte Carlo techniques, we constructed a simple dynamic system, consisting of a simulated ball bearing rolling across a nonlinear landscape (see Figure 6). In this system, the bearing has a state

$$S = [x\ \dot{x}\ \ddot{x}\ y\ \dot{y}\ \ddot{y}]$$

- Given a state $S = [s_1, s_2 \ldots s_N]$ at $t = 0$, and assuming Gaussian noise, produce

  $$a_1 = S + N(0, \Sigma_S) \ldots a_N = S + N(0, \Sigma_S),$$

  to get a vector

  $$A_{t=1} = [a_1, a_2 \ldots a_n],$$

  where $N(\mu, \Sigma)$ denotes a normally distributed random variable with mean μ and covariance matrix Σ^2. Here, Σ_s is the simulation noise covariance.

- Then, for each t until $t = T$, calculate

  $$A_{t+1} = f(A_t) + N(0, \Sigma_m(A_t)),$$

  where f is the model function and Σ_m is the model noise.

- Each element $a_1 \ldots a_N$ of $A_t = T$ is then mapped to a grain.

Figure 7. Monte Carlo time series prediction used in our ball bearing example.

The height component in the simulation isn't included because the bearing can't leave the surface. Here we assume Gaussian noise about the current state. The landscape model is also considered to be uncertain, in this case with a spatially varying uncertainty.

In Figure 6, the dark-colored grid on top of the lighter solid surface shows the two-standard deviation bound on the uncertainty; the uncertainty is Gaussian in this example and so is fully specified by its mean and standard deviation.

We can predict the distribution of positions where the ball bearing might roll, by simulating perturbations around the current state, producing N perturbed samples. Increasing this number results in a more accurate representation of the target distribution—the set of the bearing's potential future states—but at a cost of increased computational load. The model simulation is then applied to these samples for each time step, until the process reaches $t_n = t + T$, where T is a predefined time horizon (see Figure 7 for the complete algorithm).

In the ball bearing example, normal ranges of the parameters are 20–40 for N and 30–80 for T. Appropriate values of the time horizon depend on the integration constant in the simulation process and on the users' response time. Users can browse the space of future distributions by directly controlling the time horizon T. We implemented this system with an InterTrax head tracker, which allows continuous control of the time horizon with simple head movements.

In the ball bearing prediction example, control actions are assumed to be constant for the prediction's duration. Other models, such as return to zero, can easily be incorporated.

Uncertainty in the model is, in this case, simulated by adding Gaussian noise to the surface height at each time step, thus diffusing the samples in regions of high uncertainty. A more realistic but computationally intensive approach would be to draw realizations of the potential landscapes from a process with reasonable smoothness constraints, such as a Gaussian process,[13] drawing one realization for each of the sample paths. This would ensure that the predicted trajectories would have appropriate dynamics. The audio output proceeds as we described earlier, except that each grain maps directly to one sample at the time horizon. This gives an audio display of the density at time $t = T$. We could extend this to include more sophisticated models of potential user behavior by predicting likely future control actions and applying these as the simulation progresses. This requires a method for feeding back the control actions that lead to the final state.

Application domain

Our proposed display method is suitable for any continuous-control system where uncertainty exists, assuming that there's also a reasonable system model that's amenable to Monte Carlo sampling. The quality of the predictions, and therefore of the feedback, depends completely on the accuracy of the system model and the user model.

By augmenting the display with prediction, whether in the form of a complex time-series prediction or basic projection along derivatives, we can mask latencies in interfaces. Such augmentation lets us produce more responsive systems, and because the bound on acceptable latencies for auditory feedback is low (around 100–200 ms is the maximum before serious performance degradation occurs[14]), this can be a significant advantage. However, this is only feasible in cases where a reasonable interface model is known or can be learned. In the worst case, a poor and overconfident system model can lead to feedback that actively hinders the user's ability to control the system. However, if the model makes poor mean predictions, but has an appropriate level of uncertainty in these predictions, it will still benefit the user—so long as the uncertainty is dis-

played appropriately, as we've described. The more accurate the model becomes, the more useful the feedback will be.

Similarly, to provide sophisticated feedback of real-time recognition processes, we can sonify particle and condensation filters,[15] with each particle in the filter being mapped to a single audio grain. Such filters are widely used in tracking and recognition tasks. For example, the particle filtering gesture recognition system described by Michael Black et al.[16] is ideally suited to a granular auditory display. The distribution over potential models maps to the distribution over waveforms, and the distributions over phase and velocity map to distributions over time inside those waveforms. Such a display completely and accurately represents the uncertainty present in the recognition system.

Implementation example: Helicopter flight simulation

As a concrete example to apply these ideas in challenging control situations, we've integrated the predictive sonification into a commercial helicopter flight simulation package. We've used Laminar Research's X-Plane (http://www.x-plane.com/) as the underlying engine because of its sophisticated flight models for rotary-wing aircraft and the availability of an extensive API for interfacing with the simulator in real time.

Helicopter flight is a well-studied example of an interaction task that's known to be challenging. Controlling the aircraft is difficult for several reasons: Pilots must coordinate controls with four degrees of freedom; there's significant lag between input and aircraft response; and the aircraft is dynamically unstable (that is, it will not tend to return to a steady state and must be continuously controlled to remain stable).

The helicopter has certain states that are desirable (such as hovering or forward flight) and which form a small subset of all the possible motions the vehicle can make. Representing the helicopter state as a point in some state space, we can define a set of goals (from the pilot's perspective) as regions in the helicopter state space. For example, the helicopter might be represented as $\mathbf{x} = [u\ v\ w\ p\ q\ r\ \phi\ \theta]^T$, where ϕ and θ are the roll and pitch; *u, v,* and *w* are the linear velocities; and *p, q,* and *r* are angular velocities. In this representation, $\mathbf{x} = 0$ corresponds to perfect level hover.

We can define a priori densities over such a state space corresponding to the desirable regions of flight. It's then possible to apply the previously described Monte Carlo propagation and sonification technique. A suitable approximation to the system dynamics can be obtained by taking local linearizations around the current state. We can perform this by perturbing the state of the aircraft and measuring the response, to obtain matrices *A* and *B* such that $\dot{x} = A\mathbf{x} + Bu$, where *u* is the control input (lateral cyclic, longitudinal cyclic, collective and antitorque, $[\theta_0\ \theta_{ls}\ \theta_{lc}\ \theta_{0t}]^T$). See Gareth Padfield[17] for a more detailed explanation of these equations. As in the ball-bearing case above, it is assumed that *u* remains constant throughout the prediction phase.

Setting the time horizon to around the response time of a particular aircraft produces an effective sonification of potential future states. This is only possible because we've suitably represented the model's uncertainty, as the linearization-based predictions are only accurate within a small region of the state space. Other techniques that do not represent the uncertainty would produce feedback, which may lead to overconfident control and subsequent instability.

In our implementation, we created a number of straightforward goals and applied the Monte Carlo propagation/sonification algorithm to obtain samples at the time horizon. Helicopter control tasks can be divided into a control hierarchy, so that each high-level goal is composed of some subgoals, each of which we've sonified independently. For example, we split *hover* into these subgoals: level, zero velocity, and zero rotation. In this case, we chose the goal sounds arbitrarily.

This simulation demonstrates that designers can easily apply the techniques we've described to complex systems in a straightforward manner. The implementation of this simulation example requires little modification from the earlier example we described, despite the implementation's significant complexity. Any interactive system where some type of Monte Carlo sampling can be applied can be sonified in this manner.

Conclusions

We've presented a flexible technique for the sonification of user interactions with a system, which has great potential for the enhancement of human–computer interaction systems and is widely applicable. It is especially suited to continuous control contexts. There is much scope for further research in the design of models suitable for sonification via granular synthesis; the quality of the feedback depends on the quality of the modeling. **MM**

Acknowledgments

Both authors are grateful for support from the Engineering and Physical Science Research Council's *Audioclouds: Three-Dimensional Auditory and Gestural Interfaces for Mobile and Wearable Computers,* GR/R98105/01. Murray-Smith acknowledges the support of the *Multi-Agent Control* Research Training Network, European Commission Training and Mobility of Researchers grant HPRN-CT-1999-00107, Science Foundation Ireland grant 00/PI.1/C067, and the Science Foundation Ireland Basic Research Grants project *Continuous Gestural Interaction with Mobile Devices*.

The research in this article was originally published in the *Proceedings of the International Workshop on Interactive Sonification* in January 2004. The article has been revised and updated to fit with *IEEE MultiMedia's* standards.

References

1. J. Mankoff, *An Architecture and Interaction Techniques for Handling Ambiguity in Recognition-Based Input*, doctoral dissertation, Dept. Computing Science, Georgia Inst. of Technology, 2001.
2. J. Mankoff, S.E. Hudson, and G.D. Abowd, "Interaction Techniques for Ambiguity Resolution in Recognition-Based Interfaces," *Proc. 13th Ann. ACM Symp. User Interface Software and Technology,* 2000, ACM Press, pp. 11-20.
3. G. Doherty and M. Massink, "Continuous Interaction and Human Control," *Proc. European Conf. Human Decision Making and Manual Control*, Group D Publications, Loughborough, UK, 1999.
4. I. Xenakis, *Formalized Music: Thought and Mathematics in Composition*, Indiana Univ. Press, 1971.
5. C. Roads, "Granular Synthesis of Sounds," *Computer Music J.*, vol. 2, no. 2, 1978, pp. 61-68.
6. B. Truax, "Real-Time Granular Synthesis with a Digital Signal Processor," *Computer Music J.*, vol. 12, no. 2, 1988, pp. 14-26.
7. E. Childs, "Achorripsis: A Sonification of Probability Distributions," *Proc. 8th Int'l Conf. Auditory Display* (ICAD 02), Int'l Community for Auditory Display, 2002; http://www.icad.org.
8. C. Roads, *Microsound,* MIT Press, 2002.
9. C.R. Kelley, *Manual and Automatic Control: A Theory of Manual Control and Its Applications to Manual and to Automatic Systems*, Academic Press, 1968.
10. R. Jagacinski and J. Flach, *Control Theory for Humans: Quantitative Approaches to Modeling Performance*, L. Erlbaum Associates, 2003.
11. R.A. Hess and J.G. Peter, "Design and Evaluation of a Cockpit Display for Hovering Flight," *J. Guidance, Control and Dynamics*, vol. 13, 1989, pp. 450-457.
12. T. Hermann, M.H. Hansen, and H. Ritter, "Sonification of Markov Chain Monte Carlo Simulations," *Proc. 7th Int'l Conf. Auditory Display* (ICAD 01), Helsinki Univ. of Technology: Laboratory of Acoustics and Audio Signal Processing and the Telecommunications Software and Multimedia Laboratory, 2001, pp. 208-216.
13. C.K.I. Williams, "Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond," *Learning and Inference in Graphical Models*, M.I. Jordan, ed., Kluwer Academic Publishers, 1998, pp. 599-621.
14. I.S. MacKenzie and C. Ware, "Lag as a Determinant of Human Performance in Interactive Systems," *Proc. Int'l Conf. Human Factors in Computing Systems* (Inter-CHI 93), 1993, pp. 488-493.
15. M. Isard and A. Blake, "Condensation–Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision*, vol. 29, no. 1, 1998, pp. 5-28.
16. M.J. Black and A.D. Jepson, "A Probabilistic Framework for Matching Temporal Trajectories: Condensation-Based Recognition of Gestures and Expressions," *Proc. European Conf. Computer Vision* (ECCV 98), LNCS 1406, H. Burkhardt and B. Neumann, eds., Springer-Verlag, 1998, pp. 909-924.
17. G.D. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*, Am. Inst. of Aeronautics, 1996.

John Williamson is a research assistant and a PhD student at the University of Glasgow. His interests are in probabilistic interfaces, audio displays, and gesture recognition. He received a BSc in computing science from the University of Glasgow, UK, in 2002.

Roderick Murray-Smith is a reader in the Department of Computing Science at Glasgow University, and he is a senior researcher at the Hamilton Institute, National University of Ireland, Maynooth. His interests are in gesture recognition, mobile computing, manual control systems, Gaussian processes, and machine learning. Murray-Smith received a BEng and a PhD from the University of Strathclyde, UK.

Readers may contact John Williamson at jhw@dcs.gla.ac.uk.