ArtiSynth: A Biomechanical Simulation Platform for the Vocal Tract and Upper Airway

Sidney Fels[1], Florian Vogt[1], Kees van den Doel[2], John E. Lloyd[1], Ian Stavness[1], and Eric Vatikiotis-Bateson[3]
Technical Report TR-2006-10, Computer Science Dept., University of British Columbia

***Abstract*— We describe ArtiSynth, a 3D biomechanical simulation platform directed toward modeling the vocal tract and upper airway. It provides an open-source environment in which researchers can create and interconnect various kinds of dynamic and parametric models to form a complete integrated biomechanical system which is capable of articulatory speech synthesis. An interactive graphical Timeline runs the simulation and allows the temporal arrangement of input/output channels to control or observe properties of the model's components. Library support is available for particle-spring and rigid body systems, finite element models, and spline-based curves and surfaces. To date, these have been used to create a dynamic muscle-based model of the jaw, a deformable tongue model, a deformable airway, and a linear acoustics model, which have been connected together to form a complete vocal tract that produces speech and is drivable both by data and by dynamics.**

I. Introduction

Computer simulation of anatomical and physiological processes is becoming a popular and fruitful technique in a variety of medical application areas. Advancements in the computer graphics and animation fields have spawned a variety of schemes for creating fast and accurate physically-based simulation. In this paper, we describe ArtiSynth, a general purpose biomechanical simulation platform focused toward creating integrated 3D models of vocal tract and upper airway, including the head, tongue, face, and jaw. Effective computer modeling of these structures will have a wide range of applications in medicine, dentistry, linguistics, and speech research. Specific examples include (a) studying the physiological processes involved in human speech production with the goal of creating an articulatory speech synthesizer, (b) planning for maxillo-facial and jaw surgery, and (c) analyzing medical phenomena such as Obstructive Sleep Apnea (OSA).

Currently, many researchers are working on modeling different anatomical substructures of the vocal tract, such as the tongue, larynx, lips, and face, using both parametric and physically-based dynamic models [1], [2]. Many of these models are very complex and they are often developed independently of other structures. The complex aero-acoustical processes that involve the interaction of these anatomical elements with airflow and pressure waves and which eventually produce speech have also been studied [3], [4]. This approach has become particularly relevant recently due to the interest in developing natural looking and/or sounding talking heads [5], [6], [7], [8], [9]. Our simulator platform, ArtiSynth, allows the creation of hybrid models that integrates these diverse approaches together.

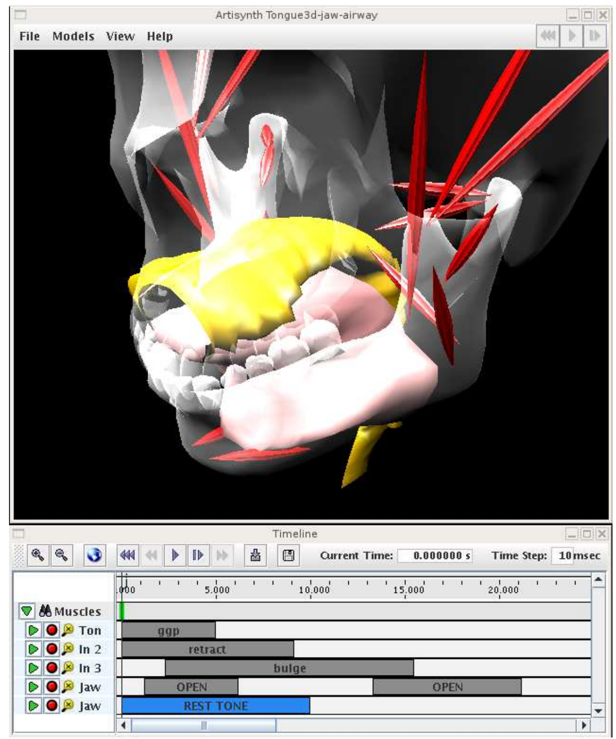

Fig. 1. ArtiSynth system, showing a partial vocal tract model (with the skull and jaw rendered transparently), along with the Timeline used for controlling it.

ArtiSynth is an open-source, cross-platform, Java-based, biomechanical simulation environment intended to

- Facilitate the creation of integrated anatomical and acoustical models by combining different kinds of dynamical and parametric sub-models;
- Provide an interactive simulation environment complete with graphical and acoustic rendering;
- Supply a basic library of model types, such as particle-spring systems, splines, and finite elements;
- Supply a baseline set of anatomical and acoustical models, combined into a complete 3D airway model, which can be used, extended, or modified by other researchers;
- Provide interactive means for users to edit models or "instrument" the simulation by attaching input/output channels to control behavior or observe properties of the model's components;

Departments of [1]Electrical and Computer Engineering, [2]Computer Science, and [3]Linguistics, University of British Columbia

A principal motivation behind ArtiSynth is the need for a platform that allows users to combine different models and modeling frameworks easily and to integrate them to form a complete biomechanical/acoustic model that can be validated both geometrically and acoustically. This will better enable the medical, linguistic, and speech communities to build on existing research and undertake collaborative projects. The present lack of such a platform is one reason why different research groups rarely integrate their models or undertake quantitative comparisons of results.

One of our specific goals is to create an open-source collaborative articulatory speech synthesizer for producing natural sounding speech. For this reason, ArtiSynth supports both anatomical and acoustical models, with the former simulating vocal tract movement and the latter simulating sound production.

Research topics that we are pursuing in the context of ArtiSynth include (1) improving the state-of-the-art in articulatory speech modeling, (2) determining accurate, efficient ways to dynamically model deformable tissue, and (3) developing techniques for using different kinds of medical imaging data in creating generic models or morphing generic models to fit individual subjects.

Thus far, we have created a core Java programming interface that implements the basic ArtiSynth system framework (described in Sec. II), along with simulation scheduling, graphic and acoustical rendering, and base classes for various dynamic and parametric models. A *Timeline* graphical user interface (GUI) (Sec. II-D) has been created to facilitate the interactive control and monitoring of the simulation. We have also created models of selected vocal tract components, including a jaw, tongue, and airway (Sec. III), and made a start at connecting these together to form a 3D articulated vocal tract that produces speech sounds. Our main focus has been on creating components that have not been developed or integrated sufficiently by other groups. Another significant part of our effort has been in developing acoustical models that can be easily connected to the anatomical models (Sec. IV).

II. ARTISYNTH SYSTEM DESCRIPTION

A. Architecture

The conceptual architecture for ArtiSynth is shown in Fig. 2. At the core is a set of *models*, along with *constraints* which control their interactions, which can be combined to form an integrated model of some physiological structure (e.g., the vocal tract). Models and constraints are implemented internally as Java classes, and are described in greater detail in Sec. II-B.

A *scheduler* is responsible for controlling the dynamic simulation of the models through time, under the control of an external GUI and *Timeline* interface (Sec. II-D). Each model is responsible for its own dynamic advancement, but can call upon various *numerics* services, such as integrators and linear system solvers.

Models may also provide methods for rendering themselves graphically or acoustically. The graphical rendering system is presently implemented using OpenGL, and permits model viewing, component selection, and various forms of graphical interaction. Acoustical rendering is described in Sec. IV.

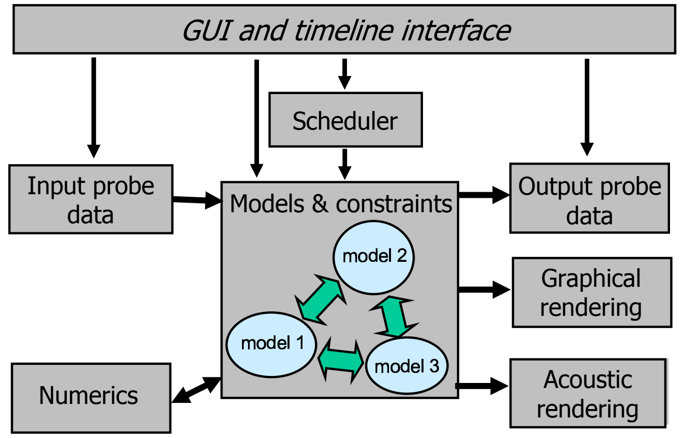

Fig. 2. General architecture of the ArtiSynth system.

ArtiSynth also provides means to "instrument" the simulation, by attaching input and output *probes* to the models or their components. Input probes are data streams which can set control inputs or modulate parameter values over time. For example, they may supply time-varying sets of activation levels to control a muscle model, or primary vocal cord waveforms to drive an aero-acoustical model. Output probes are data streams which can record model variables or properties for a portion of the simulation. For example, they may be used to log the motions of an anatomical component such as the jaw, or the final waveform produced by an aero-acoustical model. Input/output probes can be scheduled graphically by arranging icons on the Timeline (Sec. II-D). A particular model configuration, with probe settings, can be saved and restored as a *workspace*, which facilitates incremental development as well as the sharing of work with colleagues.

Fig. 1 shows a full screen shot of ArtiSynth, displaying the integrated model described in Sec. III-E and with control inputs arranged on the Timeline. Intuitively, our approach to the interface in ArtiSynth is to blend the timeline features of movie editing with that of simulation from 3D modeling tools.

B. Modeling Framework

ArtiSynth is built around the concept of a *model*, which is the key unit of simulation. Models can represent specific anatomical structures, such as a muscle-activated jaw (Sec. III-A), the tongue (Sec. III-C), the airway cavity itself (Sec. III-D), or acoustical production entities, such as the model described in Sec. IV-A). They can be implemented using ArtiSynth-supplied base classes which presently support particle/spring/rigid body systems, finite element methods of the type described by [10], spline-based curves and surfaces, and PCA-driven point clouds or meshes.

A Java class implementing an ArtiSynth model must supply two principal methods:

```
model.initialize (long t0);
model.advance (long t0, long t1);
```

The first initializes the model to time t0, and the second advances it from time t0 to t1. Both methods are used by the scheduler to drive the simulation. Models are classified as either *parametric* or *dynamic*, depending on whether

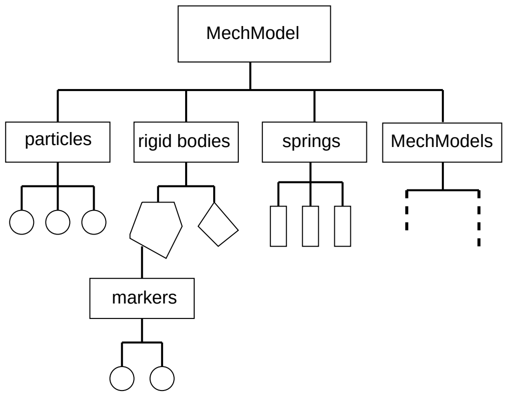

Fig. 3. Hierarchical arrangement of components in MechModel.

their advancement depends on internal *state*. Parametric models include principal component [11], [12] and spline-based systems, and are often employed to emulate the kinematic movement of deformable tissues based on medical imaging and tracking data. They are usually driven by a trajectory of parameter values. Dynamic models include rigid bodies, spring-mass and finite element models (FEMs), and are used to model the actual physics of rigid and deformable anatomical tissues [10]. They are generally driven by a trajectory of input "forces".

Because dynamic models contain state, their time advancement generally requires numerical integration from some known initial state, and there is usually a maximum step size at which this integration should be performed. Models make this step size known via the method

```
model.getMaxStepSize();
```

and the scheduler will then ensure that the step size used by `advance` never exceeds this.

Models can be arranged hierarchically, and are composed of other basic components such as particles, marker points, rigid bodies, springs, FEM elements, etc. For example, the class `MechModel` allows a dynamical system to be built out of particles, springs, rigid bodies, and other `MechModels`, as shown in Fig. 3. Rigid bodies, in turn, may contain marker points used for observation or for connections to other models. ArtiSynth provides GUI support for navigating the component hierarchy, and allows individual nodes to be named or selected. Selected nodes can be edited, and their exposed properties can be connected to input or output probes, as described in Sec. II-D.

C. Connecting Models Together

An important feature of ArtiSynth is the ability to interconnect models of disparate type (particularly dynamic and parametric models) to form a complete model of some physiological structure. This interconnection is achieved using *constraint* components, which provide a general mechanism for specifying interactions between models. Constraints enforce themselves by modifying the input and/or output variables of the models which they are interconnecting. This is done using the methods `modifyInputs` and `modifyOutputs`, which are called before and after the advancement of the relevant models. For example, for a constraint connecting `modelA` and `modelB`, the scheduler would effect the following call sequence:

```
constraint.modifyInputs (t0, t1);
modelA.advance (t0, t1);
modelB.advance (t0, t1);
constraint.modifyOutputs (t1);
```

What these `modify` methods actually do depends on the constraint being implemented. For hard constraints between dynamic mechanical models (e.g., particle connections or joints between rigid bodies) `modifyInputs` might use a Lagrangian approach to compute and apply constraint forces, while `modifyOutputs` would correct the state variables to compensate for numerical drift. For soft collision constraints, `modifyInputs` would again apply forces, but `modifyOutputs` would likely do nothing. For a constraint involving parametric models, `modifyInputs` might adjust the parametric input values, but `modifyOutputs` would again do nothing because there is no integration step in a parametric model that would require correcting.

Our most common use of constraints is to effect point-wise model connection, where points on one model are attached to points on another. Attachment points may include FEM vertices, mass particles, surface mesh vertices (for parametric models), and marker points on a rigid body. We are also implementing a non-interpenetration constraint between models, which will be implemented using soft collision forces based on interpenetration of their surface meshes.

Interconnecting models raises issues for the scheduler, which, when taking a time step, must decide which models to advance first. Consider, for example, a situation where marker points on a dynamic jaw model are attached to control points on a (parametric) spline-based airway model, whose shape in turn affects the behavior of an acoustical model. In this situation, there are constraints between the jaw and airway model, and between the airway and acoustic model. These constraints are one-way, with only one model's behavior affected by the other. Such a one-way constraint is called a *dependency*. Prior to simulation, the scheduler examines the constraints for dependency information, and uses this to build a directed acyclic graph of clusters of mutually interdependent models. This graph implies an ordering that specifies which model clusters should be advanced first. Within each cluster, models are advanced in arbitrary order, at the smallest maximum step rate for all models in that cluster. Between clusters, different step rates are possible. This allows for multi-rate integration, which is important because different time scales may be appropriate for different types of models. (This is particularly true when connecting anatomical and acoustic models, as described in Sec. IV.)

D. Interactive Simulation Control and the Timeline

A primary aim of ArtiSynth is to allow a user to interactively control the simulation, using different control inputs, and to record the resulting trajectory of specific observables. To facilitate this, ArtiSynth model components can explicitly

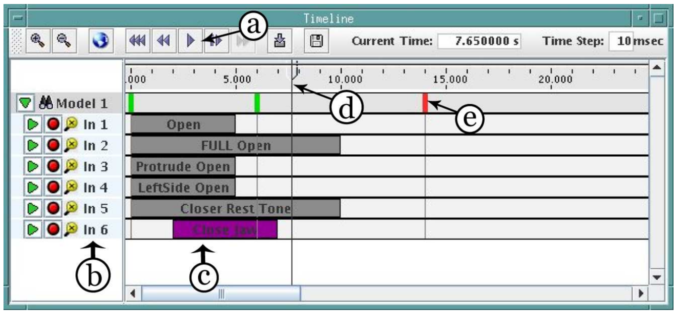

Fig. 4. The ArtiSynth Timeline, with play control buttons (a), track labels (b), probe arrangements (c), band time cursor (d). Some input probes have been arranged to control muscle activations of the jaw model. Three rectangular way points have been placed on the top track (e).

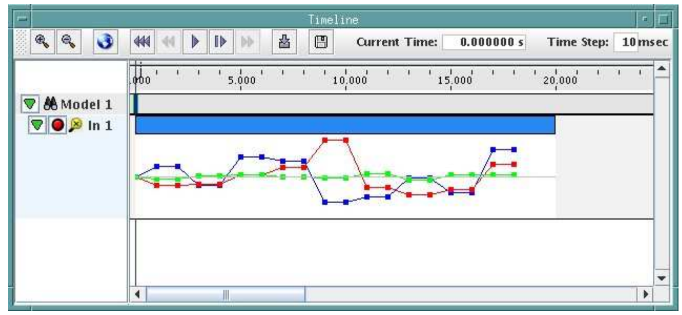

Fig. 5. Timeline with a track opened to allow editing of its probe data. The probe in question contains three linearly interpolated numeric values, which may be modified by dragging the knot points.

expose properties for reading or writing while the simulation is in progress. Examples might include the position or velocity of a particle, muscle activation levels, control points for a spline, spring stiffness, or render properties, such as line width or color. Exposed properties are defined in a generic way that makes it easy to automatically create a GUI panel for adjusting their values. (in a style similar to that used in the JavaBeans[13] framework).

Exposed properties can also be connected to input or output probes, which allow them to be continuously written or read by a data stream. Input probes typically drive the simulation and are connected to muscle activation levels, external forces, parametric model parameters, glottal excitation waveforms, etc. Output probes are connected to observable data, such as the locations of marker points or landmarks (e.g., the tongue tip), interaction forces, or generated acoustic waveforms. We also intend to extend output probes so that they can record *functions* of observed data, such as distances or angles between marker points, or cross-sectional areas of an airway mesh.

An important expected use of input probes is to explicitly supply the simulation with reference data extracted from medical image sets. An example of this would be a jaw position trajectory extracted from a set of dynamic MR images or a trajectory of tongue contours extracted from ultrasound video. We have worked on extraction algorithms for CT, MRI, and ultrasound imaging that can generate such data, and the input probe would allow it to drive a portion of the simulation or overlay the simulation with reference imagery.

Probes can be applied to a simulation at different times and for varying durations. Different input probe configurations will result in different simulation behavior and will in turn result in differing results at the output probes. To coordinate probes with the simulation, ArtiSynth provides the Timeline component (Fig. 4).

Inspired by movie edit software, the ArtiSynth Timeline is a zoomable time window with play control buttons to start, stop, and single-step the simulation. It also provides different tracks on which input and output probes may be arranged. Using mouse interaction, a user may create a probe, attach it to a specific model component, drag it to a specified start point on a track, and adjust its duration by scaling or cropping. Timeline tracks can also be “opened” to allow the viewing and editing of certain kinds of probe data (Fig. 5). When the probe arrangement is complete, the user may start the simulation using the *play* button. The current simulation time is indicated by a vertical cursor which advances left-to-right. As the cursor moves over input or output probes, driving inputs are adjusted and output data is recorded, as appropriate.

Unlike in movie edit applications, it is generally *not* possible to move the Timeline cursor to an arbitrary location. This is because the simulation typically involves dynamic models, which can only be set to a specific time by integrating from some previously known time and state. To offset this difficulty, the Timeline allows a user to set *way points* (indicated by rectangular icons), which are used to save the current model state at specific times. It then *is* possible to move the Timeline cursor between such way points (using the fast forward or reverse buttons) and restart the integration with the saved state.

E. Using Medical Image Data for Model Registration

It is important to ensure the correct alignment and registration of separate anatomical structures within the biomechanical model. To assist in this, we have utilized Amira and the Insight Toolkit, specialized medical image analysis and visualization software, to extract the necessary substructure shapes and key points by means of semi- and fully-automatic segmentation methods, such as level sets. The segmented data (images or meshes) are used to register the ArtiSynth models using rigid and non-rigid methods. Currently we have used a combination of CT, MRI, and ultrasound images to create and register the default models in ArtiSynth. Extracting accurate shapes for anatomical models, positioning rigid structures, and selecting muscle attachment sites have all been successfully accomplished through the integrated use of Amira[14] and ArtiSynth. In conjunction with the Insight Toolkit[15],custom automatic segmentation and registration have been applied in the image domain to reduce human intervention.

III. Anatomical Models

Within the ArtiSynth framework, we have produced models for the jaw, tongue, larynx, and vocal tract airway, and combined them with a linear audio model to create a hybrid three-dimensional vocal tract model capable of producing vowel sounds.

A. Jaw Models

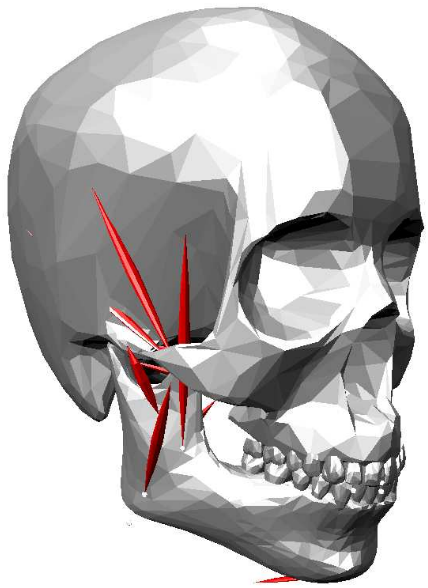

Fig. 6. The jaw model, showing the muscles (schematically represented as tapered ellipsoids) connecting the skull with the mandible.

We have implemented a three-dimensional, dynamic jaw model (Fig. 6) consisting of a fixed rigid skull, floating rigid mandible, two temporo-mandibular joints, eighteen muscles, multiple bite points, based on [16]. The rigid bodies are rendered as meshes that we extracted from CT scans. The model was developed to predict jaw-muscle tensions during simulated postural rest, jaw opening and chewing. It is driven by 18 Hill-type actuators representing nine pairs of jaw muscles [17] [18].

B. Laryngeal Model

The jaw model has been extended to include laryngeal structures including the hyoid, thyroid cartilage, and cricoid cartilage. These structures have been modeled as rigid body elements and are connected with a spring network that approximates passive connective tissue. Movement of the cricoid and thyroid is achieved with a cricothyroid joint and Hill-type actuators representing the cricothyroid muscles. Arytenoid structures and cricoarytenoid joints have been modeled with future goal of creating an anatomically accurate model of vocal chord stretching.

C. Tongue Models

We have created 2D and 3D deformable tongue models based on the quasi-linear fast FEM methods recently described by [10], [19]. The model consists of an underlying triangle (2D) or tetrahedral (3D) mesh and linear or quasi-linear stiffness matrices to specify the shape and material deformation properties. Muscle action is modeled by applying forces between specific nodes of the FEM (Fig. 7). These inter-node forces are arranged to correspond roughly to the lines of action for particular muscles, with the resulting applied forces proportional to the muscle actuation level. This tissue model is fairly general and we expect to use it for creating other deformable tissues in the vocal tract.

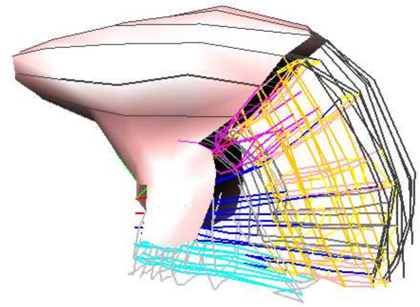

Fig. 7. A 3D finite element tongue model, showing (in the cutaway) the lines of action used to model specific muscles.

D. Airway Model

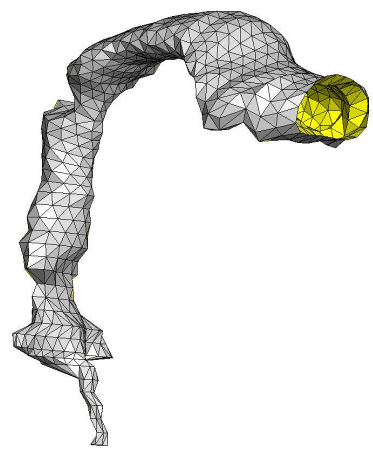

Fig. 8. The airway is modeled as a deformable mesh with additional data, such as acoustical surface impedances, which facilitate aero-acoustical modeling.

To produce speech, aero-acoustical phenomena that occur in the vocal tract airway have to be modeled. In principle, the airway is determined implicitly by its adjacent anatomical components, but as some of these components may not yet be modeled, or maybe of limited relevance, we have developed a stand-alone version of the vocal tract airway. Such explicit airway modeling is also described in [20], [21]. Our airway consists of a mesh-based surface model depicted in Fig. 8. In addition to the surface mesh, the airway model allows for additional data such as labeling particular points or areas, or surface acoustical impedances, which may be needed for acoustical modeling within the tract. The airway is deformable and can change its shape in concert with the anatomical components which surround it, as described in Sec. III-E.

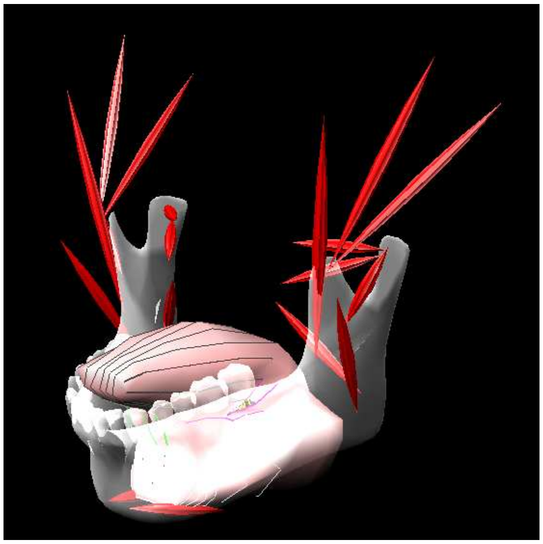

Fig. 9. The jaw and tongue models connected together (with the skull omitted for clarity).

E. Integrated Vocal Tract Model

The models described above have been connected together using the inter-model constraint mechanism (Sec. II-C). The tongue is connected to the jaw (Fig. 9) by attaching FEM nodes at the tongue base to marker locations on the mandible. A Lagrangian approach is used to compute constraint forces which keep the nodes and markers collocated and ensure consistent dynamic behavior of the coupled system.

We have also connected the airway to the jaw-tongue model (Fig. 1), to form a complete articulated vocal tract. The airway is (at present) a parametric model which must deform in concert with the surrounding structures to which it is connected. This is done by setting airway mesh vertices to correspond either directly to specific component surface vertices, *or* to a weighted combination of such vertices near gaps or junctions. This helps ensure that the airway is both topologically correct and smooth. The technique is somewhat akin to the skinning procedures used in character animation (e.g., [22]).

By further connecting the airway to the source filter audio model of Sec. IV-A, we obtain a complete articulated vocal tract capable of making vowel sounds.

F. Other Models

We have created a number of other models not described here, including a parametric principal component model of face motion [23] and a 2D vocal tract model motivated by [24] using NURBS curves. We are currently adding dynamical models of other anatomical structures, including the lips, the soft palate and uvula, and portions of the face. A stand-alone face tissue model has already been created, based on the work described in [6].

IV. Acoustical Modeling and Rendering

The motion of anatomical substructures can be modeled at a relatively coarse temporal resolution of around $50ms$. High quality audio synthesis occurs at a sampling rate of 44.1 kHz, which requires a temporal resolution of around $22ns$, about 2000 times denser. In principle the simulation capabilities of ArtiSynth can be used for the simulation of aero-acoustical phenomena as well, but this requires the entire simulation to run at the audio rate, which requires extremely long run times. Because of these widely different time scales it is often desirable to construct separate specialized models for the simulation of audio. The situation is similar to the integration of audio and motion in computer graphics. Running a detailed simulation using FEM at audio rates was attempted in [25], to calculate an animation with sound from a physical model of deformable bodies, which resulted in extremely long processing times. A different approach was taken in [26], where the audio and motion simulators are running their own specialized models in parallel at different rates. This allows for real-time high-quality interactive simulation with motion and sound.

The aero-acoustical modeling implemented to date uses the latter approach. The audio models are implemented in JASS [27], which is a cross-platform Java based real-time audio synthesis framework. JASS provides Java interfaces and abstract classes which can be extended to create unit generators, which are connected into filter-graphs, using the paradigm introduced in computer music by Max Mathews [28]. It also provides for low latency real-time audio rendering capabilities.

The audio renderer provided in ArtiSynth processes audio buffers computed by the sound model tree and either renders these in real-time for immediate feedback during a simulation, or it can write the audio data to file for later analysis. In real-time mode the audio model communicates asynchronously with the motion simulation components of ArtiSynth through specific Java interfaces which encapsulate the communication between the subsystems.

A. Audio Model

A linear audio model is currently connected to the airway model for the production of vowels. Information about the airway's 3D shape which is coupled to a subset of relevant anatomical parts, as described in section III-E, is used to generate a wave propagation channel which is excited by a glottal excitation. The glottal excitation is implemented as a special input probe (see Sec. II-D). The probe can either read PCM data from a file, or generate the glottal wave algorithmically according to the Rosenberg model [29].

The wave propagation through the vocal tract can be modeled using the well-known Kelly-Lochbaum [30] tube segment filter model, as well as more directly using the linearized Navier-Stokes equations which we solve numerically in real-time on a 1D grid using a symplectic integrator with operator splitting [31] to handle stiffness arising through small constrictions which generate strong damping. An advantage of the latter approach is that the airway can be stretched continuously (when pursing the lips for example) which is not possible with the classical Kelly-Lochbaum model which requires a fixed grid size.

Through careful use of Java interfaces, all the details concerning the airway's coupling to the surrounding anatomical structures is hidden from the audio model. This allows for easy modification to the airway model without requiring any changes in the audio code, and vice-versa. The airway model also provides an arena for more sophisticated audio modeling techniques based on fluid dynamics which we are currently developing.

The audio excitation is synchronized with the simulation of the motion to produce the integrated synthesis of motion and sound. The sound and motion can be produced in real-time, or movie- and soundtracks can be written to file.

V. Summary

ArtiSynth is an open-source platform for collaborative biomechanical modeling targeted specifically at upper airway anatomy and vocal tract acoustics. It allows researchers to create new model components, compare and predict geometric and acoustic properties of the vocal tract, and explore physiological and speech related phenomena, all within a complete system. It supplies library support the creation of a variety of dynamic and parametric models, along with baseline implementations of various anatomical components which can be easily integrated into a coherent functioning system for applications ranging from articulatory speech synthesis to medical and physiological research. A Timeline interface allows interactive temporal control of input/output probes which can drive the simulation or record observed data, and supports hypothesis testing for vocal tract articulations and acoustics based on position data or dynamic forces.

Version 2.0 of ArtiSynth was released in the fall of 2005, and an updated release is planned for the spring of 2006. New functionality currently under development includes graphical support for navigating and editing models, soft collision handling for interaction between deformable models, and simulation of airflow through the airway cavity. We continue to enhance our baseline set of anatomical components and will be adding models of the lips, soft palate, and face. Our plan is to build a significant model library of vocal tract components from expert researchers. Ultimately, by offering an open-source infrastructure for researchers to explore, modify and expand upon, we hope to encourage collaborative activity and deepen our understanding of the this highly complex part of the human anatomy. We invite researchers to participate in the project in any way possible. More information can be found at: www.artisynth.org.

Acknowledgments

This work was supported by NSERC, Peter Wall Institute for Advanced Studies and the Advanced Telecommunications Research Laboratory (Japan). We gratefully acknowledge the many contributions made from the team of people contributing to this project including: Alan Hannam, Carol Jaeger, Bryan Gick, Oliver Guenther, Ian Wilson, Rahul Chander, Justin Lam, Justine Chen, Jennifer Li, Eric Lok, and Charles Wilson. Further we would like to thank Yohan Payan, Pascal Perrier, Mark Tiede, Philip Rubin, Olav Engwall, Dimitri Terzopolous, Yuencheng Lee, Takaaki Kuratate, Maureen Stone, and Uri Ascher for contributing implementations, data, models and good advice.

References

[1] M. O. Rosa and J. Pereira, "Towards full-scale three dimensional larynx simulation," in *Proc ICVPB*, 2004.
[2] J. Dang and K. Honda, "A physiological articulatory model for simulating speech production process," *JASJ*, vol. 22, no. 6, pp. 415–425, 2001.
[3] P. Svancara, J. Horacek, and L. Pesek, "Numerical modeling of production of czech vowel /a/ based on FE model of vocal tract," in *Proc ICVPB*, 2004.
[4] D. J. Sinder, M. H. Krane, and J. L. Flanagan, "Synthesis of fricative sounds using tan aeroacoustic noise generation model," in *Proc. ASA Meet.*, June, 1998.
[5] K. Waters, "A Muscle Model for Animating Three-Dimensional Facial Expression," in *Proc SIGGRAPH*, vol. 21, 1987, pp. 17–24.
[6] Y. Lee, D. Terzopoulos, and K. Waters, "Constructing physics-based facial models of individuals," in *Proc GI*, 1993, pp. 1–8.
[7] M. M. Cohen and D. W. Massaro, *Modeling coarticulation in synthetic visual speech*. D. Thalmann N. Magnenat-Thalmann, Springer-Verlag, 1993, pp. 141–155.
[8] F. I. Parke and K. Waters, *Computer Facial Animation*. A K Peters, 1996.
[9] K. Kaehler, J. Haber, and H.-P. Seidel, "Geometry-based muscle modeling for facial animation," in *Proc GI*, 2001, pp. 37–46.
[10] M. Mueller and M. Gross, "Interactive virtual materials," in *Proc GI*, 2004, pp. 239–246.
[11] P. Badin, G. Bailly, M. Raybaudi, and C. Segebarth, "A three-dimensional linear articulatory model based on mri data," in *Proc ICSLP*, 1998, pp. 14–20.
[12] O. Engwall, "A 3d tongue model based on mri data," in *Proc ICSLP*, 2000.
[13] "Javabeans software - project homepage." [Online]. Available: http://java.sun.com/products/javabeans
[14] "Amira 3d data visualization software - project homepage." [Online]. Available: http://www.amiravis.com
[15] "Insight segmentation and registration toolkit (itk) - project homepage." [Online]. Available: http://www.itk.org
[16] G. Langenbach and A. Hannam, "The role of passive muscle tensions in a three-dimensional dynamic model of the human jaw," *Arch Oral Bio*, vol. 44, pp. 557–573, 1999.
[17] A. Hill, "The heat of shortening and the dynamic constants of muscle," *Proc Roy Soc B*, vol. 126, pp. 136–195, 1938.
[18] F. Zajac, "Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control." *Crit Rev Biomed Eng*, vol. 17, pp. 359–411, 1989.
[19] M. Nesme, M. Marchal, E. Promayon, M. Chabanas, Y. Payan, and F. Faure, "Physically realistic interactive simulation for biological soft tissues," *Recent Research Developments in Biomechanics*, vol. 2, 2005. [Online]. Available: http://www-evasion.imag.fr/Publications/2005/NMPCPF05
[20] H. C. Yehia and M. Tiede, "A parametric three-dimensional model of the vocal-tract based on mri data," in *Proc ICASSP*, 1997, pp. 1619–1625.
[21] K. Honda, H. Takemoto, T. Kitamura, and S. Fujita, "Exploring human speech production mechanisms by mri," *IEICE Info Sys*, vol. E87-D, pp. 1050–1058, 2004.
[22] N. Magnenat-Thalman, R. LaPerriere, and D. Thalman, "Joint-dependent local deformations for hand animations and object grasping," in *Proc GI*, 1988, pp. 26–33.
[23] T. Kuratate, E. Vatikiotis-Bateson, and H. C. Yehia, *Estimation and Animation of Faces using Facial Motion Mapping and a 3D Face Database*. Academic Press, in press.
[24] P. E. Rubin, T. Baer, and P. Mermelstein, "An articulatory synthesizer for perceptual research," *JASA*, vol. 70, pp. 321–328, 1981.
[25] J. F. O'Brien, P. R. Cook, and G. Essl, "Synthesizing Sounds from Physically Based Motion," in *Proc SIGGRAPH*, 2001, pp. 529–536.
[26] K. v. d. Doel, P. G. Kry, and D. K. Pai, "FoleyAutomatic: Physically-based Sound Effects for Interactive Simulation and Animation," in *Proc SIGGRAPH*, 2001, pp. 537–544.
[27] K. v. d. Doel and D. K. Pai, "JASS: A Java Audio Synthesis System for Programmers," in *Proc ICAD*, 2001.
[28] M. V. Mathews, *The Technology of Computer Music*. MIT Press, 1969.
[29] A. E. Rosenberg, "Effect of glottal pulse shape on the quality of natural vowels," *JASA*, vol. 49, pp. 583–590, 1971.
[30] K. L. Kelly and C. C. Lochbaum, "Speech Synthesis," in *Proc. Fourth ICA*, 1962.
[31] R. J. Leveque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.