Coupled hard–soft tissue simulation with contact and constraints applied to jaw–tongue–hyoid dynamics

Ian Stavness[1,†], John E. Lloyd[1], Yohan Payan[2,3] and Sidney Fels[1]

[1]*Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada*
[2]*PIMS-Europe, UMI CNRS 3069, University of British Columbia, Vancouver, Canada*
[3]*TIMC-IMAG Laboratory, UMR CNRS 5525, University Joseph Fourier, 38706 La Tronche, France*

SUMMARY

We present an open-source physical simulation system suitable for efficient modeling of anatomical structures composed of both hard and soft tissue components, interconnected by point-wise attachments, contact, and other constraints. Specific attention is paid to the computational formulation needed for the coupled simulation of rigid and deformable structures, and a constraint-based mechanism is described for attaching these together. As an application of this system, we then present a novel 3D dynamic model of the jaw–tongue–hyoid complex, consisting of an FEM model of the tongue, rigid jaw, and hyoid structures, point-to-point muscle actuators, and constraints for bite contact and the temporomandibular joints. Several simulations are presented showing combined jaw–tongue actions and demonstrating the effects of coupled jaw–tongue–hyoid dynamics.

KEY WORDS: biomechanical modeling; hard–soft tissue coupling; jaw; tongue; finite element method; interactive simulation

1. INTRODUCTION

Computer simulation is becoming an integral aspect of biomedical research and practice, in applications ranging from basic research in physiology and motor control to clinical applications in treatment planning and surgical training. An important example of this is the dynamic modeling of muscle-driven anatomical structures as a means to better understand their function in normal and pathological cases. Modeling such structures is non-trivial, and often involves the combined simulation of hard structures (such as bones), interconnected by constraints (such as joints), attached to various soft tissues (such as skin, fat, mucosa, muscle, and tendon), and in contact with each other and the environment.

One particularly challenging region to model in this regard is the human oral, pharyngeal, and laryngeal complex, pictured in Figure 1, which is critical to a wide range of important functions including chewing, swallowing, breathing, and speaking. It is composed of rigid structures including the cranium, jaw, and hyoid bone, highly deformable muscle-activated tissues such as the tongue, soft palate, and pharynx, additional deformable structures such as the larynx and articular disc, an intricate arrangement of many muscles some of which are capable of exerting very large forces, and various contact and constraint situations including bite contact and the temporomandibular joint. Given the complexity of the upper airway, it is not surprising that its function and motor

---

†Correspondence to: Ian Stavness, Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada.
 E-mail: stavness@ece.ubc.ca
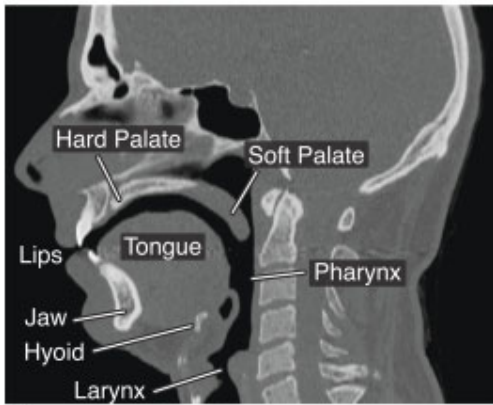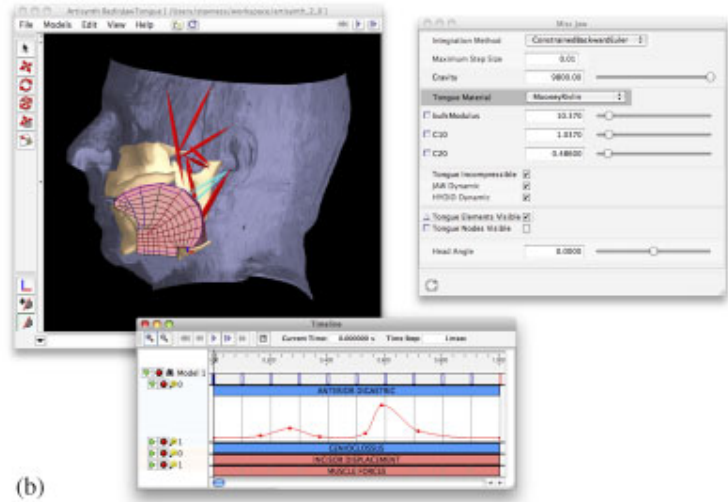
Figure 1. (a) The upper airway anatomy shown in a mid-sagittal CT image and (b) screen shot of ArtiSynth showing the jaw–tongue–hyoid model along with a timeline for controlling input/output data and a control panel for adjusting model properties.

control strategies are not completely understood, especially for complex motor actions, such as speech production, that involve the coordination of multiple structures in very fast movements up to 80 cm/s [1]. Comprehensive dynamic models of this region will enhance our understanding of both its normal physiological function and its dysfunctions, such as obstructive sleep apnea [2], swallowing disorders [3], and speech pathologies [4]. These models will also help to evaluate the consequences of maxillofacial surgical interventions [5].

Modeling complex structures for biomedical applications is enhanced by an interactive environment capable of simulating the dynamics of coupled rigid and deformable structures, together with contact and constraints, in a reasonable amount of time (seconds as opposed to minutes or hours). Traditionally, deformable structures are modeled using finite element method (FEM) approaches [6, 7] that emphasize boundary conditions and nodal loading. Rigid-body structures, on the other hand, are modeled using multibody techniques [8] that emphasize constraints and contact. Existing software packages tend to be oriented to either the FEM (e.g. ANSYS, SIMULIA) or multibody (e.g. SolidWorks, ADAMS) approach and creating hybrid models within either is generally not easy. Moreover, solution times can be very slow: a 1-s simulation of the FEM tongue model described in [9] can require many hours of computing time.

In this paper, we describe an open-source physical simulation system that combines both FEM and multibody capabilities, with an emphasis on computational efficiency. We then show the application of this system to a novel integrated model of the tongue, jaw, and hyoid complex that requires FEM/multibody capability to physically model the dynamic interactions when soft tissue and hard bodies are connected and interact. The physical simulation system is described in Section 3, along with specific discussions of attachments between bodies, contact handling, solution techniques, and a comparison of solution results with ANSYS. Our demonstration of the simulation system with a model of coupled jaw–tongue–hyoid dynamics is described in Section 4, along with a number of simulations of combined tongue–jaw actions to illustrate the types of complexities that arise in this region and how our system manages them.

Specific contributions include (1) a general mechanism for attaching rigid and deformable bodies (Section 3.5), (2) an exposition of the computational issues involved, including the need to integrate constraints directly into the velocity solution (Section 3.4), (3) a collision handling technique based on intersection contours between surface meshes (Section 3.6), (4) a simple but effective physics simulation algorithm (Section 3.7), (5) the first 3D jaw–tongue–hyoid dynamical model with full coupling between the FEM tongue model and the jaw–hyoid bony structures (Section 4.2), (6) simulations demonstrating effects of coupled jaw–tongue–hyoid dynamics, and (7) preliminary comparisons of simulations results with published human data (Section 4.3).

2. RELATED WORK

Biomechanical models of individual structures in human upper airway have been developed and used since the 1970s. Model complexity has increased due to both the acquisition of new knowledge about anatomical, neurophysiological, and physical characteristics of the articulators, and the vast growth in computational capacity. Early models were based on a 2D mid-sagittal representation of the airway [10–15], whereas recent models attend to the 3D structure of the articulators. 3D biomechanical models have been proposed for the face [5, 16, 17], jaws [18, 19], tongue [9, 20, 21], and larynx [22]. Most authors have used the FEM to discretize the partial differential equations governing the continuum mechanics of soft tissues. However, because of the significant technical challenges with properly modeling the dynamical coupling between soft bodies (tongue, lips, soft palate) and hard structures (jaw, hyoid bone, hard palate), most 3D models have proposed to focus on one single articulator. Hence, to our knowledge, our approach is unique for providing a 3D modeling framework simulating full dynamical interaction between soft and rigid articulators. Such dynamic coupling is important, especially in speech production where it has been clearly shown that consideration of the dynamic interaction of vocal tract bony and soft structures is needed to correctly account for orofacial dynamics (see [13] with the use of a quite simple 2D FEM model).

Deformable bodies often form stiff systems that require implicit integrators for computationally efficient solutions, and hence rigid–deformable coupling requires the ability to apply implicit integration techniques to constrained multibody problems. Theoretical work on this problem has been reported in [23–25]. Rigid–deformable body coupling has also been studied by the computer graphics community [26]. For surgical simulation, fast rigid–deformable contact is required for instrument interactions; methods to achieve this include penalty forces [27] and a linear complementarity approach optimized for linear materials [28].

Several open simulation systems and architectures have been presented to the biomedical community in recent years. OpenSim [29] is a multibody simulator with inverse modeling capabilities designed for musculoskeletal work, whereas FEBio [30] is a finite element toolkit with special support for tissue modeling and some support for rigid-bodies, contact and constraints. Systems geared toward surgical training include Gipsi [31] and Spring [32]. Sofa [33] provides a general software architecture in which models can be partitioned into different submodels for simulating appearance, behavior, and/or haptic response. To the best of our knowledge, none of these open simulation systems yet provide an interactive environment with fully coupled FEM/multibody capabilities.

3. COUPLED HARD–SOFT TISSUE SIMULATION WITH CONTACT AND CONSTRAINTS

3.1. The ArtiSynth system

Our physical simulation system is embedded within ArtiSynth (www.artisynth.org), an open-source Java-based biomechanical modeling toolkit developed at the University of British Columbia. Originally designed for speech applications [34], ArtiSynth has evolved into a tool for physiological research (particularly neuromotor control) and clinical treatment planning. Artisynth models are created in Java code, using the packages and classes of the ArtiSynth API. Graphical editing and model creation is also supported. Applications to date have focused on the jaw and oral region [18, 35], but it is broadly applicable to biomechanical modeling in general. Key system features include (1) an architecture that supports extensive interactivity, including graphically based model editing and simulation control (Figure 1(b)), and (2) a physics engine that combines FEM and multibody capabilities, with constraints and contact handling, as described below.

3.2. Physical simulation framework

Artisynth models consist of a hierarchy of components, which include *dynamic components* such as particles, FEM nodes, or rigid-bodies, *force effectors* such as point-to-point muscles (including Hill and other types), linear or nonlinear finite elements, and *constraints* such as joints or collision

specifiers. FEM capabilities include support for tetrahedral, hexahedral, and some higher-order elements, along with both linear and large-deformation behaviors, including corotated linear [36] and hyperelastic materials. We now describe the mathematical framework for the dynamic simulation of these models.

Let $\mathbf{q}$ and $\mathbf{u}$ be the generalized positions and velocities of all the dynamical components, with $\dot{\mathbf{q}}$ related to $\mathbf{u}$ by $\dot{\mathbf{q}} = \mathbf{Q}\mathbf{u}$ ($\mathbf{Q}$ generally equals the identity, except for components such as rigid-bodies, where it maps angular velocity onto the derivative of a unit quaternion). Let $\mathbf{f}(\mathbf{q}, \mathbf{u}, t)$ be the force produced by all the force effector components (including the finite elements), and let $\mathbf{M}$ be the (block-diagonal) composite mass matrix. By representing rigid-body velocity and acceleration in body coordinates, we can ensure that $\mathbf{M}$ is constant. Newton's second law then gives

$$\mathbf{M}\dot{\mathbf{u}} = \mathbf{f}(\mathbf{q}, \mathbf{u}, t). \tag{1}$$

In addition, bilateral and unilateral constraints give rise to locally linear constraints on $\mathbf{u}$ of the form

$$\mathbf{G}(\mathbf{q})\mathbf{u} = 0, \quad \mathbf{N}(\mathbf{q})\mathbf{u} \geqslant 0. \tag{2}$$

Bilateral constraints include rigid-body joints, FEM incompressibility associated with the mixed u-P formulation [37], and point-surface constraints, whereas unilateral constraints include contact and joint limits. Constraints give rise to constraint forces (in the directions $\mathbf{G}(\mathbf{q})^{\mathrm{T}}$ and $\mathbf{N}(\mathbf{q})^{\mathrm{T}}$) that supplement the forces of (1) in order to enforce the constraint conditions. In addition, for unilateral constraints, we have a complementarity condition in which $\mathbf{N}\mathbf{u} > 0$ implies no constraint force, and a constraint force implies $\mathbf{N}\mathbf{u} = 0$. Any given constraint usually involves only a few dynamic components and hence $\mathbf{G}$ and $\mathbf{N}$ are generally sparse. Solving the equations of motion requires integrating (1) together with (2).

The presence of deformable bodies generally makes this system stiff, implying the need for an implicit integrator to obtain efficient performance[‡]. For the work described in this paper, we use a semi-implicit second-order Newmark integrator [25], with $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$ (also known as the *trapezoidal rule*). Letting k index values at a particular time step, and h denote the time step size, this leads to the update rules

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \frac{h}{2}(\dot{\mathbf{u}}^k + \dot{\mathbf{u}}^{k+1}), \quad \mathbf{q}^{k+1} = \mathbf{q}^k + \frac{h}{2}(\mathbf{Q}^k\mathbf{u}^k + \mathbf{Q}^{k+1}\mathbf{u}^{k+1}), \tag{3}$$

subject to

$$\mathbf{G}^{k+1}\mathbf{u}^{k+1} = 0, \quad \mathbf{N}^{k+1}\mathbf{u}^{k+1} \geqslant 0. \tag{4}$$

Since $\mathbf{G}$ and $\mathbf{N}$ tend to vary slowly between time steps, we can approximate (4) using

$$\mathbf{G}^k\mathbf{u}^{k+1} = \mathbf{g}^k, \quad \mathbf{N}^k\mathbf{u}^{k+1} \geqslant \mathbf{n}^k, \tag{5}$$

where $\mathbf{g}^k \equiv -h\dot{\mathbf{G}}^k\mathbf{u}^k$ and $\mathbf{n}^k \equiv -h\dot{\mathbf{N}}^k\mathbf{u}^k$. Likewise, we use the approximation $\mathbf{Q}^{k+1} \approx \mathbf{Q}^k + h\dot{\mathbf{Q}}^k$. For $\dot{\mathbf{u}}^{k+1}$, recalling that $\mathbf{M}$ is constant, an estimate of the (unconstrained) value of $\dot{\mathbf{u}}^{k+1}$ can be obtained from $\dot{\mathbf{u}}^{k+1} \approx \mathbf{M}^{-1}\mathbf{f}^{k+1}$, with $\mathbf{f}^{k+1}$ approximated by the first-order Taylor series

$$\mathbf{f}^{k+1} \approx \mathbf{f}^k + \frac{\partial \mathbf{f}^k}{\partial \mathbf{u}}\Delta\mathbf{u} + \frac{\partial \mathbf{f}^k}{\partial \mathbf{q}}\Delta\mathbf{q}.$$

[‡]With very soft tissue, it may sometimes be possible to use explicit methods [38], particularly if stiffness-proportional damping is excluded.

Placing this into the expression for $\mathbf{u}^{k+1}$ in (3), multiplying by $\mathbf{M}$, noting that $\Delta\mathbf{q}=h/2(\mathbf{Q}^k\mathbf{u}^k+\mathbf{Q}^{k+1}\mathbf{u}^{k+1})$ and $\Delta\mathbf{u}=\mathbf{u}^{k+1}-\mathbf{u}^k$, and incorporating the constraints (5), we obtain the system

$$\begin{pmatrix}\hat{\mathbf{M}}^k & -\mathbf{G}^{k\mathrm{T}} & -\mathbf{N}^{k\mathrm{T}} \\ \mathbf{G}^k & 0 & 0 \\ \mathbf{N}^k & 0 & 0\end{pmatrix}\begin{pmatrix}\mathbf{u}^{k+1} \\ \lambda \\ \mathbf{z}\end{pmatrix}+\begin{pmatrix}-\mathbf{M}\mathbf{u}^k-h\hat{\mathbf{f}}^k \\ -\mathbf{g}^k \\ -\mathbf{n}^k\end{pmatrix}=\begin{pmatrix}0 \\ 0 \\ \mathbf{w}\end{pmatrix}, \quad 0\leqslant\mathbf{z}\perp\mathbf{w}\geqslant0. \qquad (6)$$

where $\mathbf{w}$ is a slack variable, λ and $\mathbf{z}$ give the average constraint impulses over the time step, and

$$\hat{\mathbf{M}}^k\equiv\mathbf{M}-\frac{h}{2}\frac{\partial\mathbf{f}^k}{\partial\mathbf{u}}-\frac{h^2}{4}\frac{\partial\mathbf{f}^k}{\partial\mathbf{q}}\mathbf{Q}^{k+1} \quad\text{and}\quad \hat{\mathbf{f}}^k\equiv\mathbf{f}^k-\frac{1}{2}\frac{\partial\mathbf{f}^k}{\partial\mathbf{u}}\mathbf{u}^k+\frac{h}{4}\frac{\partial\mathbf{f}^k}{\partial\mathbf{q}}\mathbf{Q}^k\mathbf{u}^k.$$

The complementarity condition for unilateral constraints is enforced by $0\leqslant\mathbf{z}\perp\mathbf{w}\geqslant0$. A more detailed explanation of this formulation can be found in [24].

System (6) is a mixed linear complementarity problem (LCP), a single solve of which is required to determine $\mathbf{u}^{k+1}$ for each semi-implicit integration step. Other types of integrators give rise to similar systems. A fully implicit integrator (not currently implemented in ArtiSynth) would require (6) to be applied iteratively at each time step.

For finite element models, the localized stiffness and damping matrices are embedded within $\partial\mathbf{f}^k/\partial\mathbf{q}$ and $\partial\mathbf{f}^k/\partial\mathbf{u}$, which means that for models dominated by FEM components $\hat{\mathbf{M}}$ will have an FEM sparsity structure. ArtiSynth FEMs also use a lumped mass model, which ensures that $\mathbf{M}$ is block-diagonal and makes it easier to interconnect FEMs with mass-spring and rigid-body components.

3.3. Friction, damping, and stabilization

Coulomb (dry) friction can be included by extending (6) to include either a linearized friction cone [23, 24] or a (more approximate but easier to solve) box friction [39]. ArtiSynth currently implements box friction, and since the friction in our system tends to be quite small, we apply this as a *post hoc* correction to $\mathbf{u}^{k+1}$ (in the manner of [26]), using a simplified version of (6), with $\mathbf{M}$ instead of $\hat{\mathbf{M}}$ and extra constraints added in the tangential directions at contact points.

Different forms of viscous damping are available, including translational and rotary damping applied directly to particles and rigid-bodies, and damping terms embedded in point-to-point springs and muscle actuators. For FEM models, Rayleigh damping is available, which takes the form

$$\mathbf{D}_\mathrm{F}=\alpha\mathbf{M}_\mathrm{F}+\beta\mathbf{K}_\mathrm{F},$$

where $\mathbf{M}_\mathrm{F}$ is the portion of the (lumped) mass matrix associated with the FEM nodes and $\mathbf{K}_\mathrm{F}$ is the (instantaneous) FEM stiffness matrix. $\mathbf{D}_\mathrm{F}$ is then embedded within the overall system matrix $\partial\mathbf{f}/\partial\mathbf{u}$.

In addition to solving for velocities, it is also necessary to correct positions to account for drift from the constraints, including interpenetrations arising from contact. This can be done at each time step using a modified form of (6), which computes an impulse $\delta\mathbf{q}$ that corrects the positions while honoring the constraints:

$$\begin{pmatrix}\hat{\mathbf{M}}^k & -\mathbf{G}^{k\mathrm{T}} & -\mathbf{N}^{k\mathrm{T}} \\ \mathbf{G}^k & 0 & 0 \\ \mathbf{N}^k & 0 & 0\end{pmatrix}\begin{pmatrix}\delta\mathbf{q} \\ \lambda \\ \mathbf{z}\end{pmatrix}+\begin{pmatrix}0 \\ \delta_g \\ \delta_n\end{pmatrix}=\begin{pmatrix}0 \\ 0 \\ \mathbf{w}\end{pmatrix}, \quad 0\leqslant\mathbf{z}\perp\mathbf{w}\geqslant0, \qquad (7)$$

where δ_g and δ_n are the constraint displacements that must be corrected. If the corrections are sufficiently small, it is often permissible to use $\mathbf{M}$ in place of $\hat{\mathbf{M}}^k$, which improves solution efficiency since $\mathbf{M}$ is constant and block-diagonal.

While such stabilization can sometimes be incorporated directly into (6) [40], we prefer to perform the position correction separately as this (a) allows for the possibility of an iterative
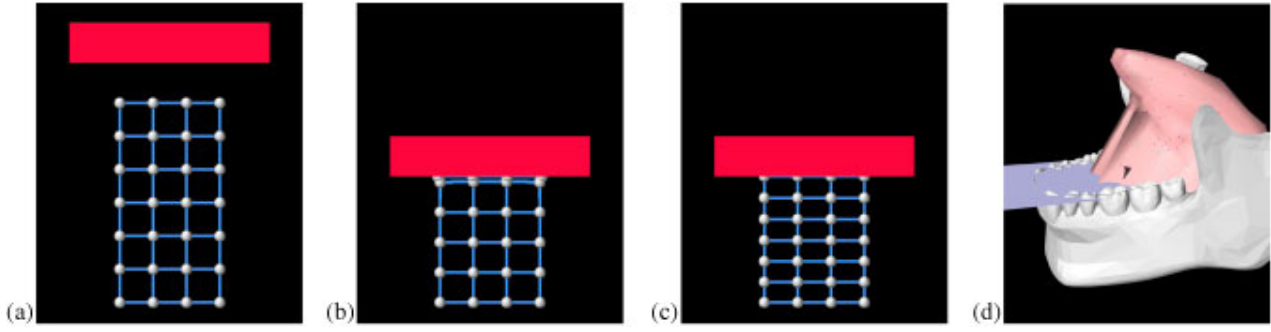
Figure 2. Problems with decoupling constraints from the velocity solve: in (a), a uniform $3 \times 6 \times 3$ FEM grid of linear material with a Poisson ratio of 0 is about to be compressed by a block. The decoupled solve causes the top contacting nodes to bunch up on the surface (b), completely squashing the top two element layers, while the lower nodes hardly move at all; the coupled solve (c) causes the correct uniform displacement for all nodes. In (d), a decoupled solve causes a tongue model attached to a jaw to exhibit large vertical errors when the jaw clenches upwards against the bite plane.

correction in the case of larger errors, and (b) explicitly separates the computed velocities from the impulses used to correct errors.

3.4. System solution and complexity

For notational convenience, in this section we will drop the k superscripts from $\hat{\mathbf{M}}$, $\mathbf{G}$, $\mathbf{N}$, $\mathbf{g}$, $\mathbf{n}$, and $\hat{\mathbf{f}}$ in (6) and assume that these quantities are all evaluated at time step k.

System (6) is a large, sparse mixed LCP [41] that is not particularly easy to solve, given the unilateral constraints and the fact that $\hat{\mathbf{M}}$ is not block-diagonal. If $\hat{\mathbf{M}}$ is symmetric positive definite (SPD), it is equivalent to a convex quadratic program. If there are no unilateral constraints ($\mathbf{N} = \emptyset$), then it reduces to a linear Karush–Kuhn–Tucker (KKT) system.

Generally, $\hat{\mathbf{M}}$ is symmetric (unsymmetric terms sometimes arise from rotational effects, but these are usually small enough to ignore) and hence will also be SPD for small enough h (since $\mathbf{M}$ is SPD). However, the resulting system is still harder to solve than non-stiff multibody systems where $\hat{\mathbf{M}} = \mathbf{M}$. This is because $\hat{\mathbf{M}}$, while still sparse, is not block-diagonal. Multi-body systems are often solved using the projected Gauss–Seidel method [39]. However, this involves a sequence of iterations, each requiring the computation of $\mathbf{G}_i \hat{\mathbf{M}}^{-1} \mathbf{G}_i^\mathrm{T}$ or $\mathbf{N}_i \hat{\mathbf{M}}^{-1} \mathbf{N}_i^\mathrm{T}$, which is easy to do for a block-diagonal $\mathbf{M}$ but much more costly for $\hat{\mathbf{M}}$.

It is tempting to follow the approach we use for friction and decouple the velocity and constraint solves, by first computing $\mathbf{u}^* = \hat{\mathbf{M}}^{-1}(\mathbf{M}\mathbf{u}^k + h\hat{\mathbf{f}})$ and then applying constraints to $\mathbf{u}^*$ in a *post hoc* fashion, using a version of (6) in which $\hat{\mathbf{M}}$ is replaced with $\mathbf{M}$. This can be done by various methods, including the Gauss–Seidel iteration, and is equivalent to projecting $\mathbf{u}^*$ onto the space of legal velocities. Unfortunately, this does not propagate constraint effects properly throughout the system, and can result in very large errors when the constraint forces are large, as illustrated in Figure 2.

At present, ArtiSynth solves (6) by using a Schur complement to turn it into a dense regular LCP

$$\bar{\mathbf{N}}\mathbf{A}^{-1}\bar{\mathbf{N}}^\mathrm{T}\mathbf{z} + \bar{\mathbf{N}}\mathbf{A}^{-1}\mathbf{b} - \mathbf{n} = \mathbf{w}, \quad 0 \leqslant \mathbf{z} \perp \mathbf{w} \geqslant 0, \tag{8}$$

where

$$\mathbf{A} \equiv \begin{pmatrix} \hat{\mathbf{M}} & -\mathbf{G}^\mathrm{T} \\ \mathbf{G} & 0 \end{pmatrix}, \quad \bar{\mathbf{N}} \equiv (\mathbf{N}\ 0), \quad \mathbf{b} \equiv \begin{pmatrix} \mathbf{M}\mathbf{u}^k + h\hat{\mathbf{f}} \\ \mathbf{g} \end{pmatrix}$$
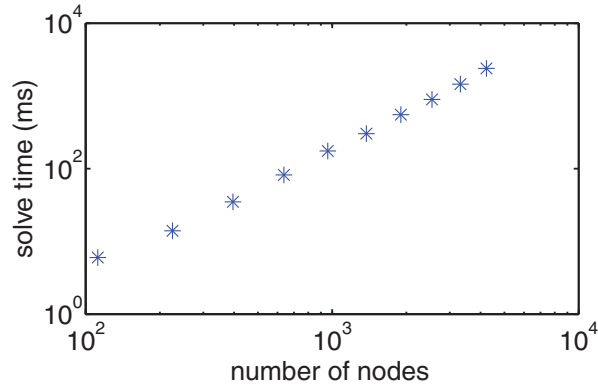
Figure 3. Log–log plot showing factor times for **A** as a function of the number of nodes (which is proportional to the size of **A**) for a series of 3D FEM problems with a uniformly increasing node density. The slope of the line indicates a complexity of $O(n^{1.7})$.

which is solved using Keller's algorithm [39]. $\mathbf{u}^{k+1}$ and λ can then be obtained using back-substitution:

$$\begin{pmatrix} \mathbf{u}^{k+1} \\ \lambda \end{pmatrix} = \mathbf{A}^{-1}(\mathbf{b} + \bar{\mathbf{N}}^{\mathrm{T}}\mathbf{z}). \tag{9}$$

Keller's algorithm is a pivoting method with an expected complexity of $O(m^3)$, where m is the number of unilateral constraints. In addition, forming (8) and the back solve of (9) requires $m+1$ solves of a system involving **A**. This is done using the Pardiso sparse direct solver [42], and entails a once-per-step factoring of **A**, plus $m+1$ solve operations. Experimentally, we have determined that the complexity of factoring **A** (using Pardiso) for 3D FEM type problems is roughly $O(n^{1.7})$, where n is the size of **A** (Figure 3). Similarly, we have also determined that the complexity of solving a factored **A** is roughly $O(n^{1.3})$. Hence, we can expect the overall complexity for solving (6) to be

$$O(m^3) + m\,O(n^{1.3}) + O(n^{1.7}).$$

This works well provided that the number of unilateral constraints m is small. To help achieve this, we can sometimes treat the unilateral constraints arising from contact as bilateral constraints (i.e. entries in **G**) on a per-step basis, as described further in Section 3.6.

Specific solution times for our jaw–tongue–hyoid model, along with a discussion of possibly more efficient solution methods, are presented in Section 5.

3.5. Attachments between bodies

In creating comprehensive anatomical models, it is often necessary to attach various bodies together. Most typically, this is done by connecting points of one body to specific locations on another body. For example, FEM nodes may be attached to particular spots on a rigid body, or to other nodes of a different FEM model.

To facilitate this, ArtiSynth provides the ability to *attach* a dynamic component to one or more *master* components. Let the set of attached components be denoted by β, and the remaining set of unattached *active* components be denoted by α. In general, the velocity $\mathbf{u}_j$ of an attached component is related to the velocities $\mathbf{u}_\alpha$ of the active components by a locally linear velocity constraint of the form

$$\mathbf{u}_j + \mathbf{G}_{j\alpha}\mathbf{u}_\alpha = 0.$$

$\mathbf{G}_{j\alpha}$ will be sparse except for entries corresponding to the master components to which j is attached. Letting $\mathbf{G}_{\beta\alpha}$ denote the composite matrix formed from $\mathbf{G}_{j\alpha}$ for all attached components, we have

$$\mathbf{I}\mathbf{u}_\beta + \mathbf{G}_{\beta\alpha}\mathbf{u}_\alpha = 0$$

for the constraints that enforce all attachments.

We could simply add these constraints to (6) and solve the resulting system, but this would increase both the system size and solution time. Instead, we use the attachments to actually reduce the size of (6). Consider first the subsystem involving only bilateral constraints. As in Section 3.4, we drop the k superscripts from $\hat{\mathbf{M}}$, $\mathbf{G}$, $\mathbf{N}$, $\mathbf{g}$, $\mathbf{n}$, and $\hat{\mathbf{f}}$ in (6) and assume that these quantities are all evaluated at time step k. Letting $\mathbf{b} \equiv \mathbf{M}\mathbf{u}^k + h\hat{\mathbf{f}}$ and partitioning the system into active and attached components yields

$$
\begin{pmatrix}
\hat{\mathbf{M}}_{\alpha\alpha} & \hat{\mathbf{M}}_{\alpha\beta} & \mathbf{G}_{\alpha\alpha}^T & \mathbf{G}_{\beta\alpha}^T \\
\hat{\mathbf{M}}_{\beta\alpha} & \hat{\mathbf{M}}_{\beta\beta} & \mathbf{G}_{\alpha\beta}^T & \mathbf{I} \\
\mathbf{G}_{\alpha\alpha} & \mathbf{G}_{\alpha\beta} & 0 & 0 \\
\mathbf{G}_{\beta\alpha} & \mathbf{I} & 0 & 0
\end{pmatrix}
\begin{pmatrix}
\mathbf{u}_{\alpha}^{k+1} \\
\mathbf{u}_{\beta}^{k+1} \\
\boldsymbol{\lambda}_{\alpha} \\
\boldsymbol{\lambda}_{\beta}
\end{pmatrix}
=
\begin{pmatrix}
\mathbf{b}_{\alpha} \\
\mathbf{b}_{\beta} \\
\mathbf{g}_{\alpha} \\
0
\end{pmatrix}.
$$

The identity submatrices make it easy to solve for $\mathbf{u}_{\beta}^{k+1}$ and $\boldsymbol{\lambda}_{\beta}$:

$$
\mathbf{u}_{\beta}^{k+1} = -\mathbf{G}_{\beta\alpha}\mathbf{u}_{\alpha}^{k+1}, \quad \boldsymbol{\lambda}_{\beta} = \mathbf{b}_{\beta} - \hat{\mathbf{M}}_{\beta\alpha}\mathbf{u}_{\alpha}^{k+1} + \hat{\mathbf{M}}_{\beta\beta}\mathbf{G}_{\beta\alpha}\mathbf{u}_{\alpha}^{k+1} - \mathbf{G}_{\alpha\beta}^T\boldsymbol{\lambda}_{\alpha}
$$

and hence reduce the system to

$$
\begin{pmatrix}
\hat{\mathbf{M}}' & \mathbf{G}'^T \\
\mathbf{G}' & 0
\end{pmatrix}
\begin{pmatrix}
\mathbf{u}_{\alpha}^{k+1} \\
\boldsymbol{\lambda}_{\alpha}
\end{pmatrix}
=
\begin{pmatrix}
\mathbf{b}' \\
\mathbf{g}_{\alpha}
\end{pmatrix},
\tag{10}
$$

where

$$
\hat{\mathbf{M}}' \equiv \mathbf{P}\hat{\mathbf{M}}\mathbf{P}^T, \quad \mathbf{G}' \equiv \mathbf{G}\mathbf{P}^T, \quad \mathbf{b}' \equiv \mathbf{P}\mathbf{b} \quad \text{with } \mathbf{P} \equiv (\mathbf{I} \quad -\mathbf{G}_{\beta\alpha}^T).
$$

Similarly, unilateral constraints can be reduced via $\mathbf{N}' = \mathbf{N}\mathbf{P}^T$. The reduction operation can be performed in $O(n)$ time and results in a system that is less sparse but generally faster to solve than the original.

Most attachments in ArtiSynth are point-based, with the most common kind being the attachment of an FEM node to a rigid body. It is also possible to attach FEM nodes to the faces and edges of an FEM element, allowing us to handle the so-called 'T-junction' problem and create FEM models with non-conforming element faces. This is quite useful for creating localized subdivisions of particular elements, particularly hexahedrons.

3.6. Contact handling

Collision detection can be enabled between any combination of rigid or deformable bodies. It is assumed that the bodies in question contain a triangular surface mesh. A bounding-box hierarchy is used to determine if any two surfaces' meshes intersect. If they do, then a tracing algorithm (similar to [43]) is used to compute all the intersection contours between the two meshes as shown in Figure 4. Such contour tracing can be done relatively quickly but does require the use of robust geometry predicates similar to those in [44]; this is particularly true because collision conditions tend to drive the contacting surfaces into degenerate mesh configurations.

Determining the intersection contour allows us to easily create a set of constraints for correcting the interpenetration and preventing interpenetrating velocities. It also provides a good estimate of the contact area, which can be used for determining contact pressure.

For collisions involving a deformable body, we locate all mesh vertices that are interior to the contour. Each such vertex corresponds to surface FEM node, which is interpenetrating the other body. For each interpenetrating node, we then find the nearest point and face on the opposite mesh, and use the face's normal $\mathbf{n}$ as a contact normal. A linear one-dimensional constraint is then created that prohibits relative motion in the negative normal direction between the node and nearest point. If the opposite face is located on a deformable body, this results in a constraint
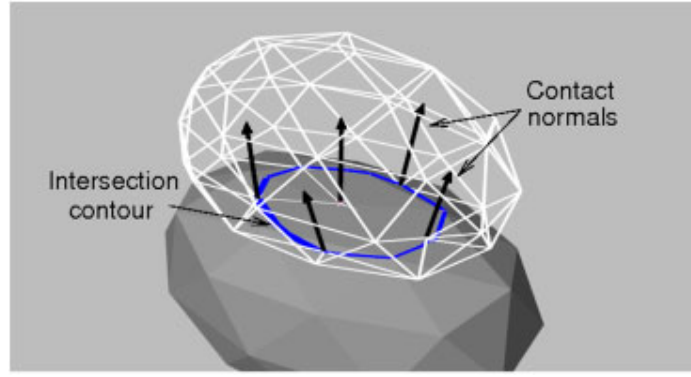
Figure 4. Contact handling between two deformable models (with the topmost rendered as a wireframe), showing the intersection contour (blue) and the contact normals (black lines) of interpenetrating vertices from the upper mesh.

between the velocity of the node $\mathbf{v}_n$ and the velocities $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$ of three nodes associated with the nearest face:

$$\mathbf{n}^\mathrm{T}\mathbf{v}_n - w_0\mathbf{n}^\mathrm{T}\mathbf{v}_0 - w_1\mathbf{n}^\mathrm{T}\mathbf{v}_1 - w_2\mathbf{n}^\mathrm{T}\mathbf{v}_2 \geqslant 0,$$

where w_0, w_1, and w_2 are the barycentric coordinates of the nearest point with respect to the face. If the opposite face is located on a rigid body, then the constraint is between $\mathbf{v}_n$ and the body's translational and angular velocities, $\mathbf{v}_b$ and $\boldsymbol{\omega}_b$ (expressed in body coordinates):

$$\mathbf{n}^\mathrm{T}\mathbf{v}_n - \mathbf{n}^\mathrm{T}\mathbf{R}\mathbf{v}_b + \mathbf{n}^\mathrm{T}\mathbf{R}(\mathbf{p}\times\boldsymbol{\omega}_b) \geqslant 0,$$

where $\mathbf{R}$ is the rotational transformation from body to world coordinates and $\mathbf{p}$ is the location of the nearest point in body coordinates. Each of these constraints can be expressed in the general form $\mathbf{N}_i\mathbf{u} \geqslant 0$, where $\mathbf{N}_i$ is a row of $\mathbf{N}$ and $\mathbf{u}$ is the vector of all velocities.

These constraints serve both to prevent interpenetrating velocities in the contact direction, and to remove interpenetrations during the position correction step (7), with the correction distance taken to be the distance d between the interpenetrating node and its nearest point on the other body.

Intersection contours are also used to determine contact constraints for rigid-body/rigid-body contact, although we omit the details here for brevity.

As mentioned in Section 3.4, the solution time of (6) can be greatly improved if some contact constraints can be temporarily treated as bilateral constraints within a particular time step. By default, ArtiSynth does this for contact involving deformable bodies, since such bodies have many degrees-of-freedom and their contact constraints tend to be somewhat decoupled. To prevent sticking, each contact's vertex–face pair is stored between time steps, and if it reappears in the next step, it is used as a contact constraint only if its corresponding λ value computed in (6) is greater than or equal to 0, implying that there is no force trying to make it separate. This is in effect an active set method, with the active set used to solve (6) being updated between steps.

It should be noted that we do not claim that the collision handling scheme described here is optimal for all applications. In particular, we do not currently implement edge–edge type contacts, and so there can be some interpenetration that depends on the coarseness of the surface meshes. However, the collision handling is properly isolated from the rest of the simulation, and other collision handling schemes can be easily used as long as they provide a set of constraints for enforcing the contact and resolving interpenetrations.

3.7. Simulation engine summary

The complete ArtiSynth simulation engine is summarized below. It uses the concept of [26, 45] whereby velocities are computed in advance of positions, subject to constraints, to help prevent constraint violations during the subsequent position computation.

1. Compute contacts (as per Section 3.6) and the bilateral and unilateral constraint matrices $\mathbf{G}^k$ and $\mathbf{N}^k$.
2. Correct positions $\mathbf{q}^k$ to remove interpenetration and drift errors, using (7).
3. If necessary, adjust $\mathbf{G}^k$ and $\mathbf{N}^k$ to reflect changes in $\mathbf{q}$.
4. Solve for $\mathbf{u}^{k+1}$ using (6).
5. Adjust velocities $\mathbf{u}^{k+1}$ for dry friction, as described in Section 3.3.
6. Compute new positions: $\mathbf{q}^{k+1} = \mathbf{q}^k + h/2(\mathbf{Q}^{k+1}\mathbf{u}^{k+1} + \mathbf{Q}^k\mathbf{u}^k)$.

This algorithm is generally applicable to any rigid-deformable body dynamics. In Section 4, we demonstrate that it can be effectively used for simulating the combined dynamics of the jaw, tongue, and hyoid bone.

In the absence of constraints, the above system turns into a trapezoidal rule solution of a regular ODE, for which global errors can be expected to be proportional to $O(h^2)$. The inclusion of constraints, particularly non-smooth unilateral ones, makes formal convergence and error analysis more difficult. However, the main velocity update (6) is the same as that described in [24], which is shown to be stable and has second-order convergence under certain assumptions.

Generally speaking, our method is a *time-stepping* scheme that uses fixed (or adapatively varying) time steps, as opposed to an *event-driven* scheme in which the integration time intervals are precisely aligned with contact events but which becomes impractical in the presence of large numbers of contacts. To the extent to which results exist, time-stepping schemes are typically shown to have less accuracy but better convergence properties than event-driven ones [46].

3.8. Validation using ANSYS

To help assess the performance of our integration scheme, we compared it against the commercial finite element package ANSYS for two test examples: a beam, fixed at one end and allowed to fall under gravity, and a cube, resting on a flat surface and subjected to a downward load applied to several top nodes. It should be noted that ArtiSynth uses several simplifications compared with ANSYS, notably the use of semi-implicit integration and a lumped mass model.

The beam example (Figure 5(a)) consisted of a beam with dimensions $0.1 \times 0.02 \times 0.02$ m divided uniformly into $8 \times 4 \times 4$ hexahedral elements, with a density of 1040 kg/m^3, Rayleigh damping coefficients of $\alpha = 20$/s and $\beta = 0.015$ s, and a five-parameter Mooney Rivlin material with $c_{10} = 10\,370$, $c_{20} = 486$ and $c_{01} = c_{11} = c_{02} = 0$ Pa. Incompressibility in both system was enforced using a mixed u-P formulation [37], and time integration was performed for 0.4 s using a 1 ms time step. To assess dynamic performance, the resulting z displacement and velocity of a reference node located in the middle of the free end were compared over time between ArtiSynth and ANSYS (Figure 6 (left)). The dynamic behavior was essentially identical: the resulting displacement error (relative to the maximum displacement) had maximum and average values of 0.3 and 0.08%. Likewise, the resulting velocity error (relative to the maximum velocity) had maximum and average values of 1.3 and 0.4%. We also determined the errors in total displacement and Von Mises stress for all the nodes in the final position: the maximum and average displacement errors (relative to the maximum displacement) were 0.06 and 0.04%, whereas the maximum and average Von Mises stress errors (relative to the maximum stress value) were 0.9 and 0.13%.

The cube example (Figure 5 (b)) used a cube with a width of 0.1 m in all directions and divided uniformly into $6 \times 6 \times 6$ hexahedral elements, with a density of 1040 kg/m^3, Rayleigh damping coefficients of $\alpha = 20$/s and $\beta = 0.015$ s, and a five-parameter Mooney Rivlin material with $c_{10} = 1037$, $c_{20} = 486$, and $c_{01} = c_{11} = c_{02} = 0$ Pa (identical to the material used for our tongue model described below). In addition to gravity, an immediate external load of -0.8 N was applied in the vertical direction to the nine nodes located in the middle of the top surface, resulting in the deformation shown in Figure 5. Incompressibility in both system was enforced using the B-bar method [37], and the example was integrated for 0.2 s with a 1 ms time step. To assess dynamic performance, a reference node was selected in the middle of the top surface and its z displacements and velocities were compared (Figure 6 (right)). Displacement errors had maximum and average values of 2.7 and 1.5%, whereas the velocity errors had maximum and average
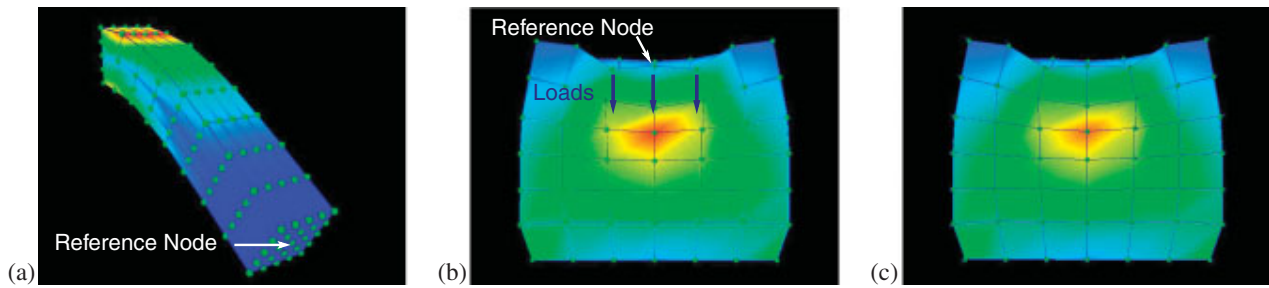
Figure 5. Examples used for validation, shown in their final positions with a rainbow plot of the resulting Von Mises stresses and the locations of their respective reference nodes. (a) Beam example, ArtiSynth; (b) cube example, ArtiSynth (maximum stress 2787 Pa); and (c) cube example, ANSYS (maximum stress 2661 Pa).
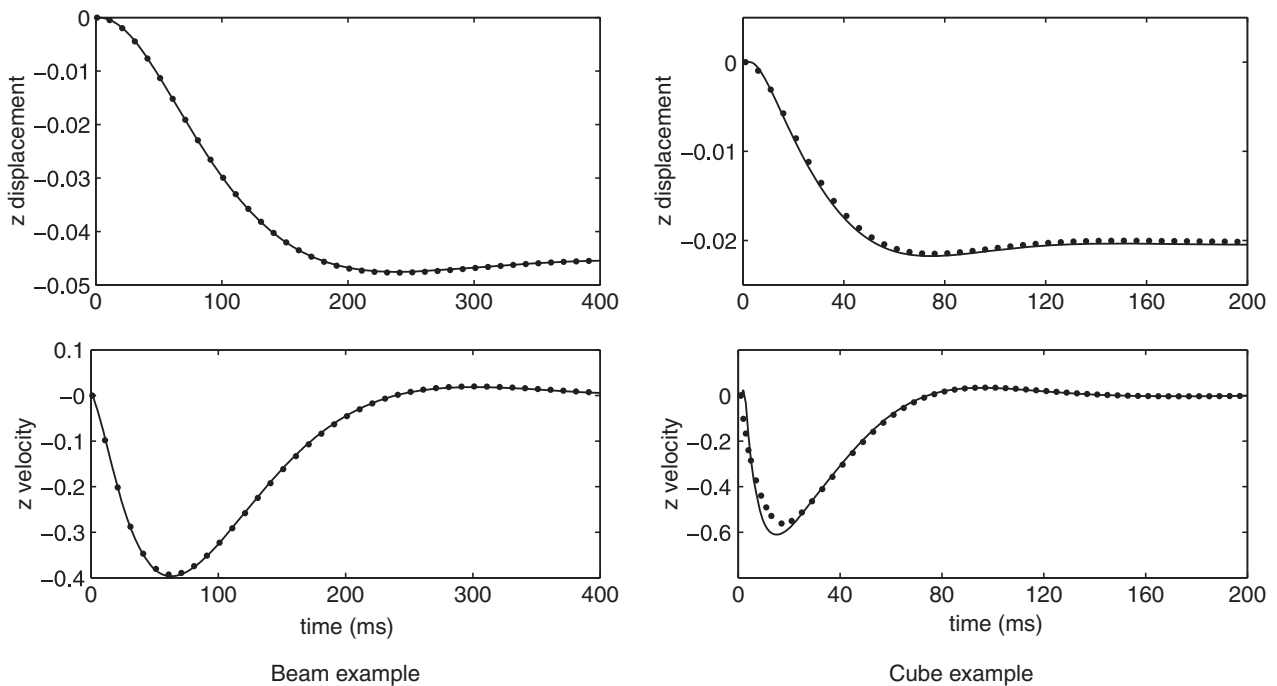

Figure 6. Time integration comparisons between ANSYS (solid lines) and ArtiSynth (dotted lines), showing the z displacement (top) and velocity (bottom) of a single reference point in the beam and cube examples.

values of 22 and 1.4%. The large value for the velocity error occurred at the beginning, where ANSYS computed an unexpected initial upward velocity for the node. Compared with ANSYS, the ArtiSynth behavior was slightly more damped. For all nodes in the final position, the maximum and average displacement errors were 3.5 and 0.5%, whereas the maximum and average Von Mises stress errors were 5.6 and 0.5%. Much of this error was due to differences in the way ArtiSynth and ANSYS compute pressure for the B-bar method, resulting in different dilational displacements: in ArtiSynth the model compressed slightly, whereas in ANSYS it inflated slightly.

These results help to demonstrate that our simulation approach is competitive with commercially available codes, in addition to be considerably more efficient: ArtiSynth was 20 and 10 times faster for the beam and cube examples, respectively.

4. APPLICATION TO JAW–TONGUE DYNAMICS

Building on our experience with the 3D jaw–hyoid [18] and the 3D tongue [9] models, we introduce here the first 3D jaw–tongue–hyoid dynamical model taking into account full coupling between the FEM tongue model and the jaw–hyoid bony structures. As described in Section 3, we depend

on the various components of Artisynth to provide dynamic simulations of interactions due to muscle forces of the jaw–tongue–hyoid complex and contact phenomena such as tongue–palate collisions.

A desire to better understand structure–function relationships and motor control strategies in upper airway movement has led a number of scientists to develop biomechanical models of individual structures in the human oral, pharyngeal, and laryngeal complex. For example, in the context of upper airway pathophysiology, biomechanical modeling enables an analysis of the structural causes of observed dysfunction. Biomedical applications typically require patient-specific models of the corresponding structures (jaw, lips, tongue, soft palate, larynx, and pharynx). Upper airway modeling has been applied to obstructive sleep apnea [2, 47] and dysphagia (disorders of swallowing) [3], as well as planning and evaluating the consequences of maxillofacial surgery [5, 48], segmental jaw surgery [35, 49], and glossectomy [4]. Further, in the context of speech production, modeling the physics of speech articulators (jaw, lips, tongue, soft palate, and larynx) enables an evaluation of the motor control strategies underlying the production of speech gestures. Physical speech signals are indeed the consequence of the interaction between the mechanical structure of the articulatory apparatus and the forces that act on it. Part of the forces involved are centrally controlled, while other forces depend on interactions between articulators and on peripheral feedback. Thus, to study the control of speech production from the articulatory or acoustic signals, researchers have proposed to model separately the biomechanics of the articulators and the neurophysiology of the control [11, 15, 50].

Section 4.1 describes the jaw–tongue–hyoid model implemented in Artisynth, whereas Sections 4.2 and 4.3 report simulations of simple lingual and mandibular motor tasks, focusing on the dynamical interaction between the tongue soft structure and the jaw rigid body.

4.1. Jaw–tongue–hyoid model

For the dynamic tongue model, we implemented the published model by Buchaillard *et al.* [9] in ArtiSynth, as pictured in Figure 7. The original tongue model was based on the anatomy of a single subject using CT data and developed in the ANSYS environment representing the tongue with hexahedral finite elements and hyperelastic properties. Thanks to a collaboration with Buchaillard and colleagues, we were able to obtain data for the 3D tongue mesh and description of the lingual muscular fibers. The mesh and muscle geometry were imported into the ArtiSynth environment, using a large-deformation FEM framework, hexahedral elements with a density of $1040 \, \text{kg/m}^3$, and a fifth-order incompressible Mooney–Rivlin material with $c_{10} = 1037$, $c_{20} = 486$, and $c_{01} = c_{11} = c_{02} = 0 \, \text{Pa}$. The deviatoric potential energy $\hat{\Psi}$ of this material is hence

$$\hat{\Psi} = c_{10}(\bar{I}_C - 3) + c_{20}(\bar{I}_C - 3)^2,$$

where $\bar{I}_C$ is the first invariant of the deviatoric component on the left Cauchy–Green tensor [7]. Incompressibility was implemented using a constraint-based mixed u-P formulation (as discussed in Section 3.2). Rayleigh damping coefficients of $\alpha = 40 \, \text{s}^{-1}$ and $\beta = 0.03 \, \text{s}$ were used to provide critical damping within the range of modal frequency from 3 to 10 Hz, consistent with the reference tongue model [9].

Muscles are represented by sets of elements and implemented with node-to-node fiber forces distributed throughout the muscle elements along the principal direction of action. While the material is isotropic, the muscle fiber components provided an approximation of the transversely anisotropic nature of skeletal muscle. We chose to use a straightforward model for muscle activation with fiber forces directly scaled by input activation, as opposed to the λ-model used in the original tongue model. Our aim was to quantify the coupling between tongue and jaw and not to work on the λ-model motor control assumptions provided by the equilibrium point hypothesis [51]. Tissue stiffening due to muscle activation was also modeled in the same way as Buchaillard and colleagues, i.e. a linear increase of c_{10} and c_{20} values ranging between (1037 Pa, 486 Pa) at no activation and (10370 Pa, 4860 Pa) at full muscle activation. As in the original model, each muscle's force capacity was a function of its cross-sectional area (see Table I in [9]), with force capacity distributed across fibers weighed by the volume of their surrounding elements.
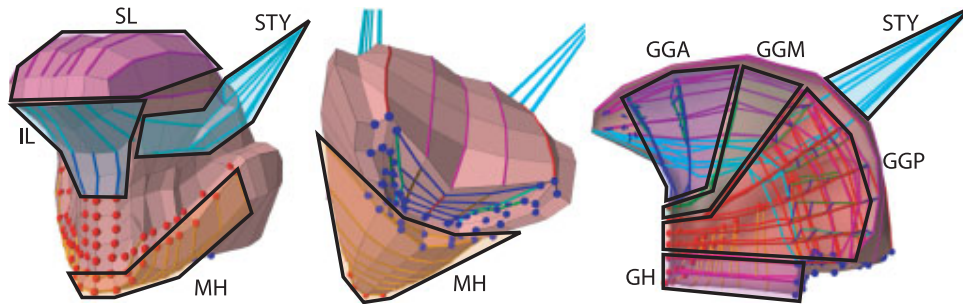
Figure 7. Front, back, and sagittal cutaway views of tongue model. Attachment nodes are also shown for the jaw (front view, red spheres) and the hyoid bone (back view, blue spheres). Muscle groups include Genioglossus (GGA, blue; GGM, green; GGP, red), Styloglossus (STY, cyan), Geniohyoid (GH, magenta), Mylohyoid (MH, orange), Hyoglossus, (HG, red), Vertical (VERT, green), Transverse (TRANS, blue), Inferior Longitudinal (IL, cyan), and Superior Longitudinal (SL, magenta).

Table I. Percentage muscle activation used in jaw–tongue–hyoid tasks.

	Jaw tasks			
	CLR*	AD	ILP	SLP
Rest	—	—	—	—
Clench	10	—	—	—
pen	—	15	15	15
Hinge-open	—	15	—	—
Protrude	—	—	15	15
Right-lateral	—	—	15[†]	15[†]

	Tongue Tasks					
	CLR*	SL	TRANS	GGP	GGM	STY
Retract	—	—	—	—	—	25
Palate	0.5	30	30	60	30	—
Max-palate	1	100	100	80	30	10

*Jaw closing muscles (AT, MT, PT, MP, SM, and DM).
[†]Only left-sided muscles are activated.

For the jaw–hyoid model, we started from a previously published model developed in ArtiSynth that has been used to simulate free jaw movements [52] and chewing [18]. The model included rigid-bodies for the skull, jaw, and hyoid bone, point-to-point Hill-type actuators for the jaw muscles, constraint surfaces for the temporomandibular joints, and planar unilateral constraints for teeth contact. To conform the disparate morphology of the two models, we adapted skeletal and muscle geometry of jaw–hyoid model to fit CT data (shown in Figure 1(a)) for the subject on whom the Buchaillard tongue model was based. The 3D jaw, skull, and hyoid surface meshes were morphed with a non-elastic mesh-based registration algorithm [53] to a 3D skull surface segmented from CT data. Symmetry was attained by mirroring the left-side of the registered meshes. The inertia of the jaw and hyoid were computed from new mesh shapes, assuming uniform density of 3600 and 2000 kg/m^3 for the jaw and hyoid, respectively. Jaw muscle origin and insertions points were adapted with the same non-elastic transformation as was applied to the surface meshes and were manually verified to correspond to plausible anatomical landmarks. We removed the point-to-point geniohyoid and mylohyoid muscles from the jaw–hyoid model as these were included in the tongue model. The anterior and posterior digastric muscles were connected to the hyoid bone with the digastric sling modeled as a pulley. The same Hill-type muscle dynamics were used from the original jaw–hyoid model with force capacity proportional to maximum cross-sectional area (see Table I in [35]). The hyoid bone was attached to a fixed larynx with a linear translational/rotational spring representing the hyothyroid membrane and ligament. The spring stiffness was set to be
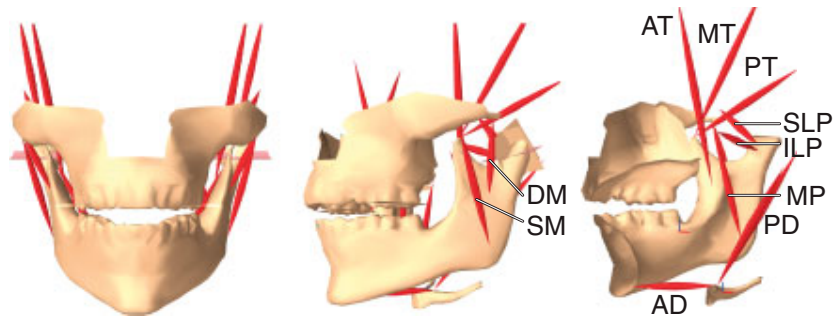
Figure 8. Front, oblique, and sagittal cut-away views of the jaw–hyoid model in ArtiSynth. Jaw–hyoid muscles modeled include the Anterior, Middle, Posterior Temporalis (AT, MT, PT), Deep and Superficial Masseter (DM, SM), Medial Pterygoid (MP), Superior and Inferior Heads of the Lateral Pterygoid (SLP, ILP), and Posterior and Anterior Bellies of Digastric (PD, AD).
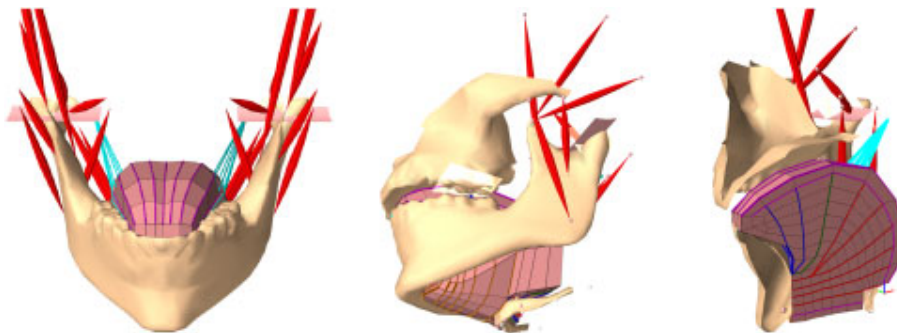

Figure 9. Front, oblique, and sagittal cutaway views of the coupled jaw–tongue–hyoid model.

consistent with the springs used to anchor the hyoid bone in the Buchaillard tongue model (eight vertical springs at 220 N/m). The adapted jaw–hyoid model is shown in Figure 8.

We couple the dynamics of the jaw, tongue, and hyoid models by defining *attachment* constraints between the FEM nodes of the tongue and the jaw and hyoid rigid-bodies, as described in Section 3.5. Point to rigid-body attachments can be made at arbitrary locations and is not required to be coincident with the rigid-body surface mesh. The attachment points in the model are shown in Figure 7. Tongue–jaw attachments include the insertion of genioglossus and geniohyoid onto mandibular geniotubercle and the insertion of mylohyoid along mandibular mylohyoid ridge. Tongue–hyoid attachments include the entire region around the anterior–superior surface of the hyoid bone, including insertions of geniohyoid, mylohyoid, and hyoglossus muscles. The posterior medial surface of tongue is not attached allowing the base of tongue to move relative to the hyoid bone. The soft palate and palatoglossus muscle are not included in the current model. The resulting combined model is pictured in Figure 9.

4.2. Simulation descriptions

We chose to simulate a set of tasks similar to those reported for the jaw and tongue models in isolation, including free jaw movements [52], unilateral chewing [18], and tongue movements in speech [9]. All of the tasks, with the exception of unilateral chewing, involve simple input muscle activations so that the passive dynamic coupling effects can be better isolated. Our objective was to analyze the effect of dynamic coupling using muscle activation to drive the coupled jaw–tongue–hyoid model and observe differences in the movement of the 'active' body as well as movement induced on the 'passive' body.

4.2.1. Jaw movement tasks. Jaw movement tasks used jaw muscle activation as input and were performed both with the jaw–hyoid model alone and with the jaw–tongue–hyoid model. All jaw tasks involved a simple pattern of muscle input (rest, ramp-up, hold, ramp-down, rest) as illustrated
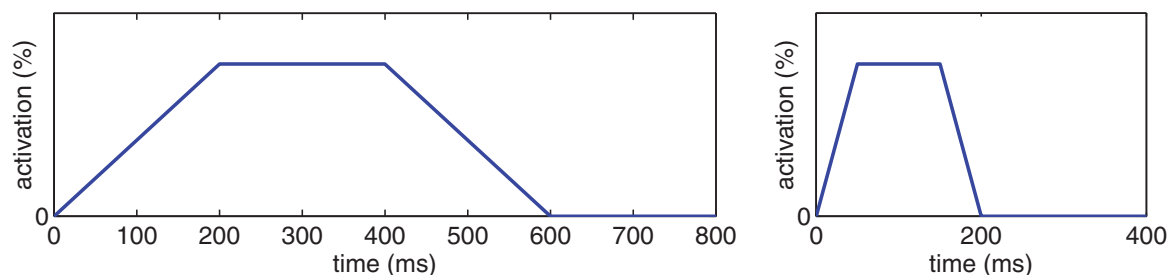
Figure 10. Input muscle activation pattern for jaw tasks (left) and tongue tasks (right). The activation amplitude for each task is given in Table I.

in Figure 10. The durations of these standardized jaw movements (600 ms) are consistent with an average chewing cycle. The muscle sets and activation amplitudes for the jaw tasks are summarized in Table I.

A nominal jaw movement task is rest posture: the equilibrium position of the model with no muscle activation and under downward gravity. In relaxed humans, the jaw typically rests with a 4–6 mm incisal separation. We expect slightly wider gape in jaw–tongue–hyoid model than in the jaw–hyoid model alone, although a majority of the tongue body rests on hyoid, and therefore should result in minimal jaw lowering. Static jaw clenching was simulated with bilateral activation of jaw closing muscles. With no tongue muscle activation, we expect the tongue to remain stationary within the mouth during tooth clenching.

Opening is simulated by bilateral activation of lateral pterygoids along with the anterior belly of digastric. We chose to simulate a moderate opening gape, with 15% activation in each muscle. Maximum jaw opening in humans is 50 mm on average although both backward head rotation and hyoid positioning become important at wide gape [54] and would complicate the task. We also simulated hinge-like jaw opening with activation of digastric alone. We expect reduced opening with jaw–tongue–hyoid model due to passive compression at floor-of-mouth. We also expect the tongue to passively protrude from the mouth during jaw opening [55], and require contact handling (see Section 3.6) between the tongue tip and the lower teeth.

Protrusion is simulated by bilateral activation of lateral pterygoid muscles. The effect of jaw protrusion on the tongue has important implications for OSA as a common therapeutic device is a dental appliance used to advance the jaw and tongue in order to open airway [56]. We expect reduced jaw protrusion in jaw–tongue–hyoid case as the lingual elastic connection between the jaw and hyoid should provide some resistance to jaw movement. Also we expect forward translation of the base of tongue. Right laterotrusion is simulated by activation of the left-side lateral pterygoid muscles. As for protrusion, we expect reduced lateral deviation in the laterotrusion task with jaw–tongue–hyoid model as compared with the jaw–hyoid model alone.

Unilateral chewing involves a complex pattern of jaw muscle activity. We simulated right-sided chewing movement using the same muscle activation patterns and food bolus that were reported for our original jaw model [18]. An elastic, spherical food bolus (10 mm in diameter) was positioned between the right first molars, which provided resistance during the closing phase of the chewing stroke and collapsed when the applied force exceeded 35 N. Since our adapted jaw–hyoid model has a different bone and muscle geometry, we expect that its chewing movement will be altered, but still plausible, as compared with the original jaw–hyoid model. We also expect that the chewing movement for the jaw–tongue–hyoid model will be significantly altered as the muscle patterns were previously tuned by Hannam *et al.* to a model without a tongue.

4.2.2. Tongue movement tasks. Tongue movement tasks used tongue muscle forces to move and deform the tongue within the mouth. Tongue tasks involved a ramp-up, hold, and ramp-down pattern of muscle input similar to the jaw tasks, but with faster transitions (50 ms, see Figure 10) for consistency with the speed of speech movements. Tongue retraction was simulated by activating styloglossus using the activation trajectory shown in Figure 10. We expect that, with the jaw at rest, a retracted tongue posture should induce backward movement of jaw.
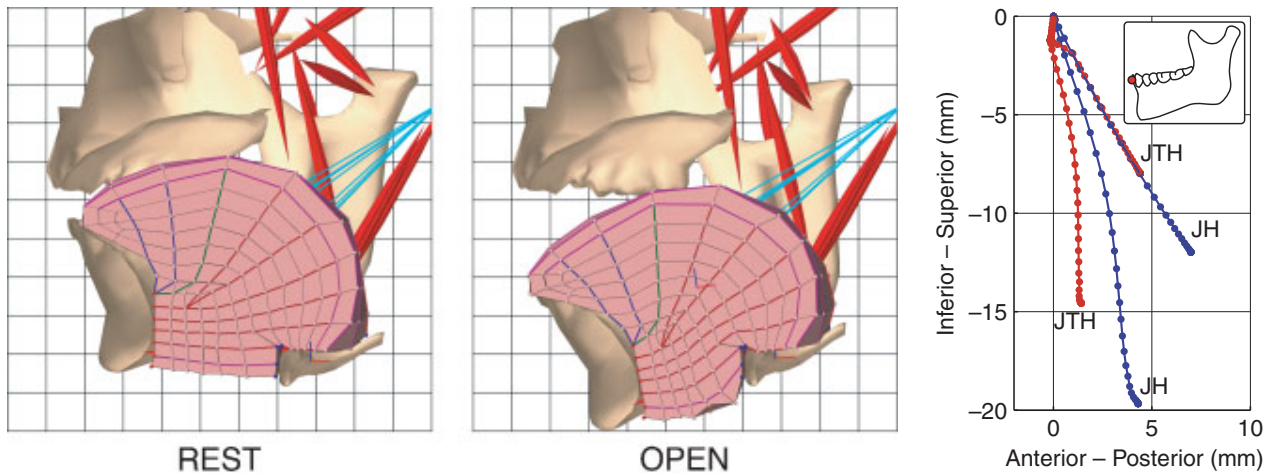
Figure 11. The jaw–tongue–hyoid model pictured during rest posture (REST) and at peak jaw opening (OPEN) with grid spacing of 10 mm. With the tongue muscles at rest, jaw opening induced tongue protrusion such that the tip rested on the lower teeth. Right-most plot shows the mid-sagittal incisor point movement for opening and hinge-opening with the jaw–hyoid model alone (JH, blue) and the jaw–tongue–hyoid model (JTH, red) with point spacing of 10 ms.

Tongue–palate contact is an important movement for speech. Tongue tip contact with the anterior hard palate was simulated by activation of superior longitudinal, posterior genioglossus, and transverse muscles (see Table I). We stabilized the jaw with low-level (0.5%) activation of jaw closing muscles to maintain a nearly closed jaw posture. We expect that tongue–palate contact will induce a downward movement on the jaw, causing the jaw to open wider. We also performed a maximum tongue–palate pressure simulation by ramping the superior longitudinal and transverse muscles to maximum activation. The ability to generate tongue-to-palate pressure is an important component of healthy swallowing function and we expect that the model's maximum tongue–palate pressure will be comparable with recorded maximum tongue pressure measurements. The tongue–palate contact simulation required contact handling between the tongue tip and the hard palate surface mesh, as discussed in Section 3.6.

4.3. Simulation results

Videos of our simulation results and information for acquiring source code for the model are available on our web site:
`http://www.artisynth.org/orofacial/ijnmbe2010/`.

4.3.1. Jaw–tongue–hyoid coupling. We observed a number of interesting influences of dynamic coupling on the simulated jaw–tongue–hyoid movements. Jaw movements were altered by the presence of the tongue and tongue movements were observed to induce jaw movement. We found a resting jaw posture with 5.6 and 6.6 mm incisal gape for the jaw–hyoid and jaw–tongue–hyoid models, respectively. Static clenching with the jaw–tongue–hyoid model simulated correctly with the tongue remaining stationary in the mouth.

The results of the jaw opening and hinge-opening simulations are shown in Figure 11. In both cases, the amplitude of jaw opening is reduced in the jaw–tongue–hyoid model due to compression of the lower portion of the tongue between the jaw and hyoid. Figure 11 also illustrates the 3D model at the wide gape position showing that jaw opening does indeed cause passive forward protrusion of the tongue such that the tongue tip is resting on the lower teeth.

Protrusion and laterotrusion movements are shown in Figure 12. The amplitude of protrusion was reduced in the jaw–tongue–hyoid model, likely due to stretching of tongue tissue between the jaw and hyoid, but the amplitude of lateral deviation was comparable. Interestingly, the tongue also induced significant downward movement of the jaw during protrusion and laterotrusion. The tongue model is pulled forward during jaw protrusion transferring more of its weight from the hyoid bone to the jaw.
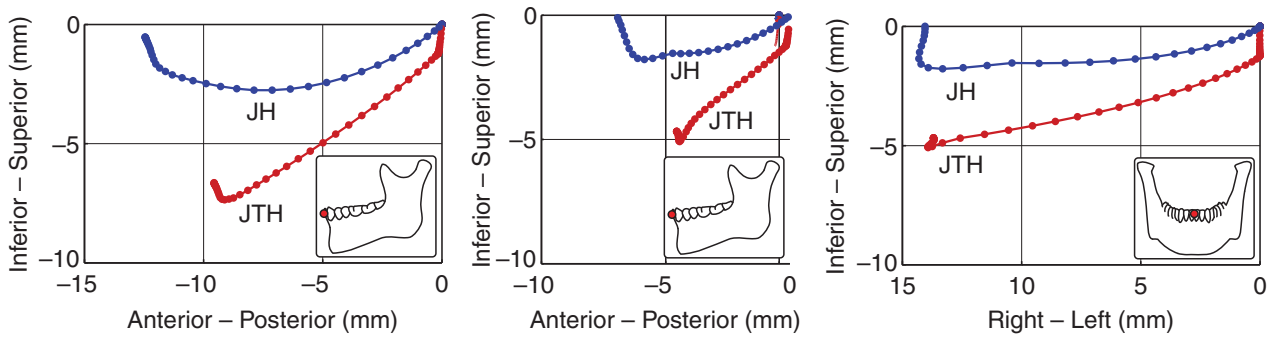
Figure 12. Incisor point movement in the sagittal and frontal planes during jaw–muscle activated protrusion and right laterotrusion for jaw–hyoid model alone (JH, blue) and the jaw–tongue–hyoid model (JTH, red) with point spacing of 10 ms.
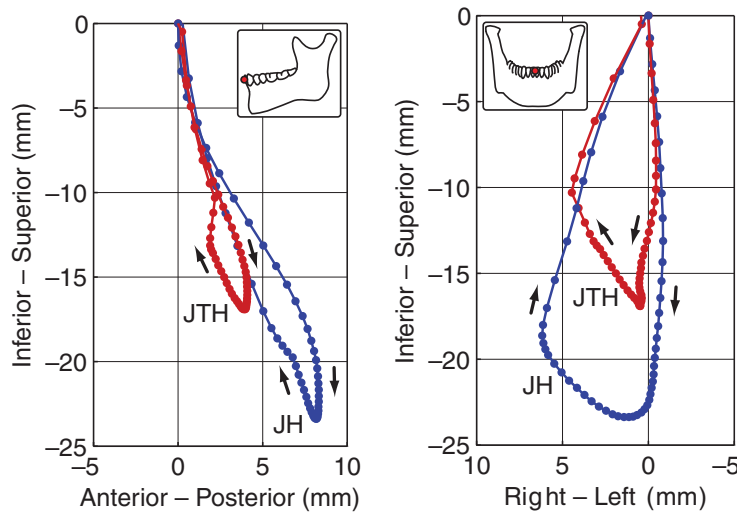

Figure 13. Incisor point movement in the sagittal and frontal planes during jaw–muscle activated right-sided chewing for jaw–hyoid model alone (JH, blue) and the jaw–tongue–hyoid model (JTH, red) with point spacing of 10 ms.

Right-side chewing movement produced by applying muscle patterns tuned for a different jaw geometry provided a plausible tear-drop-shaped incisor movement in the current jaw–hyoid model as shown in Figure 13. The movement compares well with the one produced by the original jaw–hyoid model (see Figure 2 in [18]). The amplitude of the chewing envelope is reduced with the jaw–tongue–hyoid model, which is consistent with the jaw opening and laterotrusion simulations.

Simulation of tongue retraction in the mouth is pictured in Figure 14 (upper panels), along with a plot of incisor displacement (lower left panel). Styloglossus activation initially causes an upward and backward movement of the incisor as the condyles move up the articular slope, followed by a backward and downward displacement as the tongue retracts farther. The tongue retraction simulation also demonstrates the large range of tongue movement capable with the model and motivates the need for a large-deformation FEM approach.

Figure 15 shows the mid-sagittal position of jaw, tongue, and hyoid for the tongue–palate contact simulation. Tongue lifting and palate contact causes a downward jaw movement as expected. The right-most panel plots the incisor point displacement, which starts before tongue-palate contact (as denoted by the × on the plot). The initial downward jaw movement is caused by tongue muscle activation and it increases as force is applied between the tongue and the palate.

4.3.2. Comparison with published data. The jaw–tongue–hyoid model has been assembled from previously reported reference jaw [18] and tongue models [9]. Qualitative evaluation of the coupled model shows similar levels of force and range of movement as exhibited by each individual model;
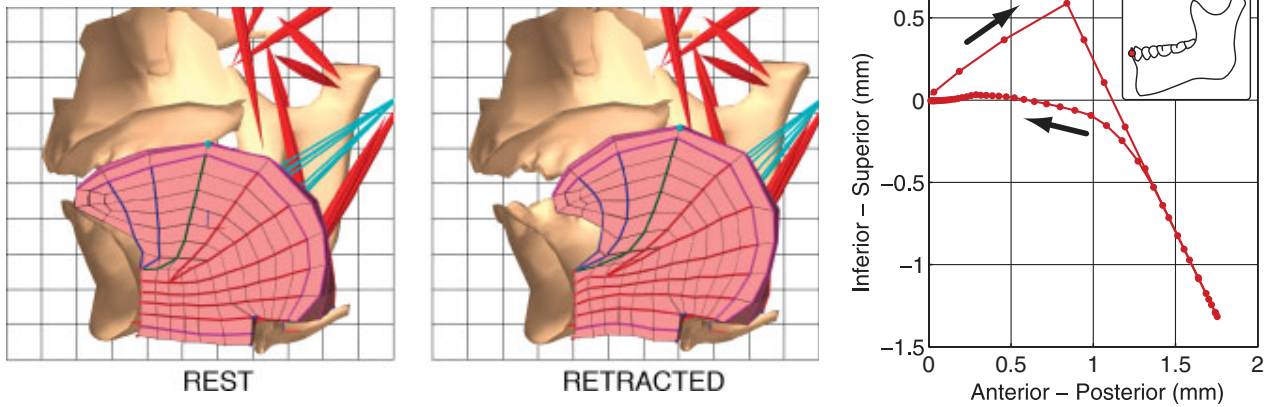
Figure 14. The jaw–tongue–hyoid model pictured during rest posture (REST) and at tongue retraction with styloglossus activation (RETRACTED) with grid spacing of 10 mm. The right-most panel plots the incisor displacement in the mid-sagittal plane with point spacing of 10 ms.
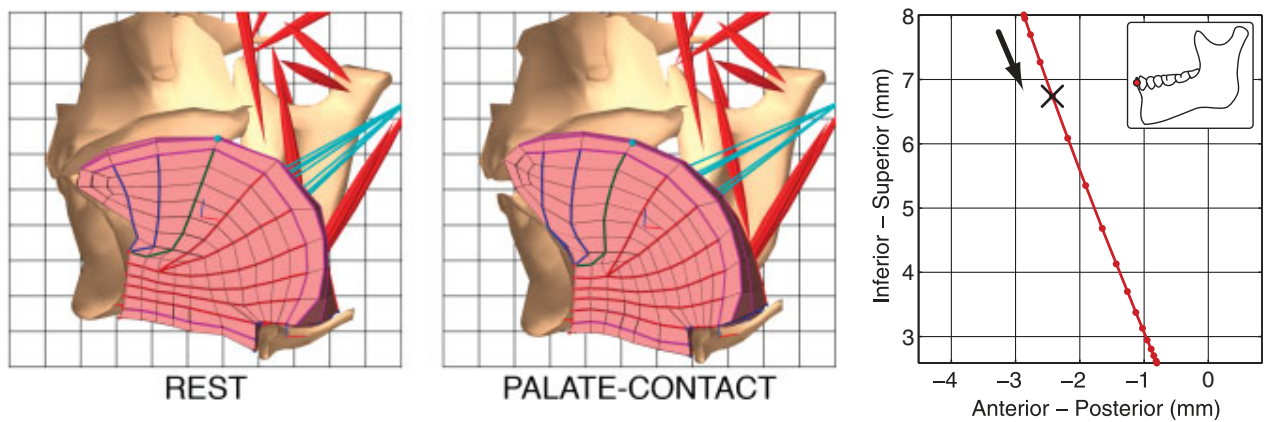

Figure 15. The jaw–tongue–hyoid model pictured during rest posture (REST) and with tongue lifted into contact with palate (PALATE-CONTACT) with grid spacing of 10 mm. The right-most panel plots the incisor point displacement, which starts before tongue–palate contact ($\times$ denotes the beginning of tongue–palate contact) with point spacing of 10 ms.

therefore, the new implementation of these models in the ArtiSynth framework compares well with the previously published versions. As a step toward the validation of the jaw–tongue–hyoid model, we have made preliminary comparisons to published data on tongue kinematics and forces.

We used the tongue retraction simulation as a means to compare tongue velocity generated in the model with measured tongue movement during speech. Payan and Perrier [12] recorded the movement of surface tongue points with an electromagnetic articulograph during tongue retraction in an [y-o] speech utterance for the speaker on whom the tongue model was based. Figure 16 shows the recorded velocity profile reproduced from [12] as well as the simulated anterior–posterior velocity of one node on the tongue's upper surface. The backward movement has a bell-shaped velocity profile with a 220 mm/s peak value and a 150 ms duration, which compares well with the tongue velocity during the recorded tongue retraction in speech.

We used the maximum tongue–palate pressure provided by the simulation as a means to evaluate whether or not the tongue muscle forces incorporated into the model are within a plausible range. Direct measurement of muscle force *in vivo* is not possible; therefore, we rely on external force measurements as an indirect means to evaluate the resultant force generation capability with the jaw–tongue model. In particular, Utanohara *et. al* [57] used a balloon-type disposable oral probe to measure tongue pressure by having subjects compress it onto the palate with maximum voluntary effort. The authors recorded pressures for a large subject pool (850 subjects) and reported 40.4 ± 9.8 kPa (mean $\pm$ standard deviation) maximum tongue pressures for subjects between 40 and 49 years of age. Pressure between the tongue–palate contact in the model was calculated by dividing
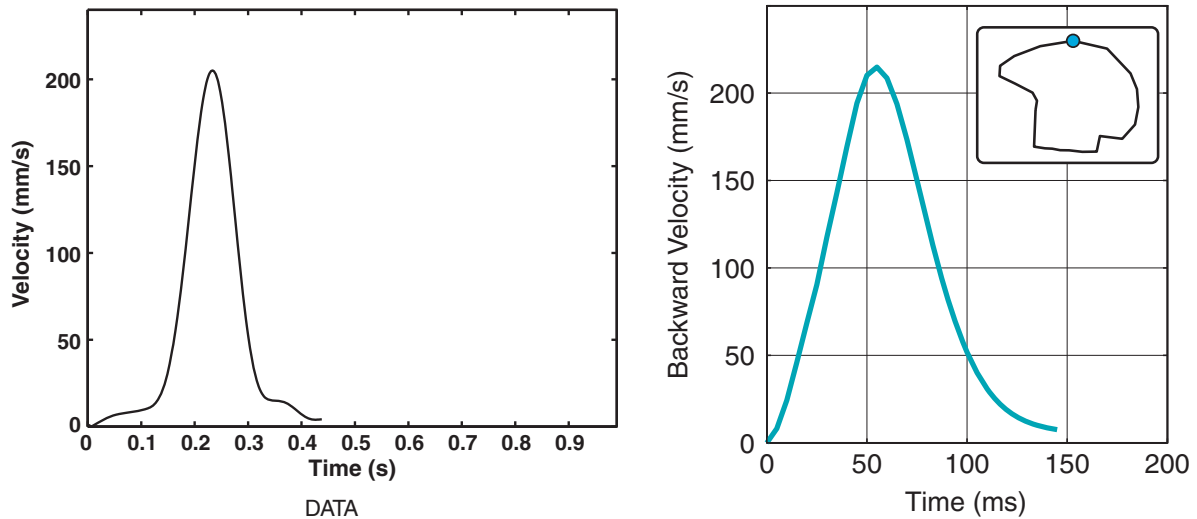
Figure 16. Backward velocity of a point on the upper tongue surface. (DATA) Backward velocity profile for a point on the upper back surface of the tongue recorded with an electromagnetic tracking system an during [y-o] speech utterance (reproduced from [12] page 15, Figure 9(a)). (MODEL) Backward velocity profile generated by the model during tongue retraction simulation.

the magnitude of the contact constraints by the area of the contact contours (see Section 3.6 for discussion of collision detection and handling in ArtiSynth). Maximum tongue–palate pressure simulated with the model was 38.2 kPa, which compares well with the values reported by Utanohara *et al.* [57].

5. DISCUSSION

5.1. Simulation results

The simulations reported here are a proof of concept demonstrating the effectiveness of our hard–soft tissue simulation and motivating the need for including dynamic coupling in simulations of jaw–tongue–hyoid movements. The reported simulations demonstrate that a wide range of movement, large forces, and large tissue deformations are possible within the current simulation framework. We have provided preliminary qualitative comparisons of tongue velocity and pressure to illustrate that the model behaves within a plausible range of human movement and force production. Further validation is planned, as additional quantitative data were recorded from the subject on which the model is based [58], although transducing upper airway movement and forces is a significant challenge.

Biomechanical simulations of the type reported here are indeed challenging to create. However, the interactivity afforded by the fast simulation times in ArtiSynth allows for reasonably fast refinement and exploration of the model's capability. Complex movements require precise coordination among a large number of muscle input degrees-of-freedom making trial-and-error tuning of muscle inputs to generate simulations tedious and likely over-fitted to a particular model's geometry. We believe that optimization-based inverse dynamics approaches are a potential direction in this regard, with which we had some experience in ArtiSynth [35].

Our preliminary results also point to a few promising directions. The results of the chewing simulation show that the muscle patterns of [18] are applicable to a different skull morphology, as they produced a very similar chewing pattern, suggesting that they are not overly sensitive to skeletal or muscle geometries. The results also show that the addition of passive tongue tissue has a significant effect on free jaw movements and the chewing movement, due to both the passive elastic connection between the jaw and hyoid made by the tongue, especially in compression, e.g. reducing jaw opening, as well as the additional mass of the tongue body, particularly in jaw protrusion.

We plan to investigate what changes in muscle patterns are required to improve the chewing stroke in the jaw–tongue–hyoid model, e.g. increasing jaw opener muscle activation. It is noteworthy that the original muscle patterns reported by Hannam *et al.* were reported as being low in amplitude; therefore, muscle activation amplitudes could be increased and remain plausible. We also plan to investigate activating the tongue muscles in concert with the jaw muscles to simulate tongue movements [59] and palate contact pressure patterns [60] during the chewing cycle. Interestingly, this would add a dynamically changing inertia as opposed to the current passive tongue mass. We are also in the process of integrating our jaw–tongue–hyoid model with an FEM model of the face [17] to analyze the effect of facial passive soft-tissue forces on jaw movement.

The tongue tissue is currently modeled as isotropic with point-to-point actuators embedded within the material to model anisotropic muscle fiber forces. We are incorporating a transverse-isotropic material [61] into ArtiSynth, which will provide a more realistic representation of skeletal muscle mechanics; however, in the current model we increase the stiffness of elements associated with muscle activation as an approximation of skeletal muscle stiffening. The tongue muscles are particularly challenging to model because multiple muscle groups, with different principal fiber directions, converge and inter-digitate within the tongue body. For this reason we are investigating a formulation allowing for the superposition of multiple transverse-isotropic materials with different principal directions.

5.2. *Integration error*

The simulations reported in Section 4.3 were computed using a second-order Newmark integrator (see Section 3.2) with a 5 ms integration step size. In order to assess the numerical error, we compared with results computed with a 1 ms integration step. Differences between 5 and 1 ms integration steps were found to be small. Figure 17 plots lower incisor point and tongue tip displacements for the jaw opening and tongue–palate contact simulations computed with 5 and 1 ms step sizes. Error was computed as the difference between the displacement trajectories relative to the maximum displacement. The jaw opening simulation showed a very small difference between the 5 and 1 ms integration step conditions: the incisor displacement error had average and maximum values of 0.5 and 1.4% and the tongue tip displacement error had average and maximum values of 0.7 and 1.6%. Larger discrepancies were found in the tongue–palate contact task, which involves significant contact situations: the incisor displacement error had average and maximum values of 1.1 and 3.8% and the tongue tip displacement error had average and maximum values of 2.0 and 10.0%. The integration error in the contact simulation is likely due to the discontinuous nature of contact as well as the spatial discretization of the palate/tongue surface meshes on which collisions are detected and responses are generated. Even in the worst case, although, the integration error remains smaller than the range of uncertainty associated with the model's assumptions and approximations.

5.3. *Computational performance*

The jaw–tongue–hyoid model described above has two free rigid-bodies and 946 FEM nodes for a total of 2505 degrees-of-freedom. With respect to (6), the addition of point-based tongue attachments, incompressibility, and jaw joints results in an $\hat{\mathbf{M}}$ that is 2505×2505 and a $\mathbf{G}$ that is typically 2505×740 (varying somewhat depending on the number of FEM contacts). In addition, a few unilateral constraints are used to implement bite contact. Solution times for (6) using the method described in Section 3.4 vary from around 130 to 200 ms (depending on whether unilateral constraints are in play) on a 2.6 GHz Core 2 Duo processor. Overall solution time (including collision detection and all the steps of Section 3.7) for a 600 ms jaw opening task with a time step size of 5 ms is around 40 s and a 400 ms tongue retraction task with a time step of 10 ms is around 20 s. Much of this involves Java code that could be significantly optimized. This improves on the computation time reported in [9], where a 100 ms task for the same FEM tongue model (with jaw/palate contact) required 40 min of computing time using ANSYS.
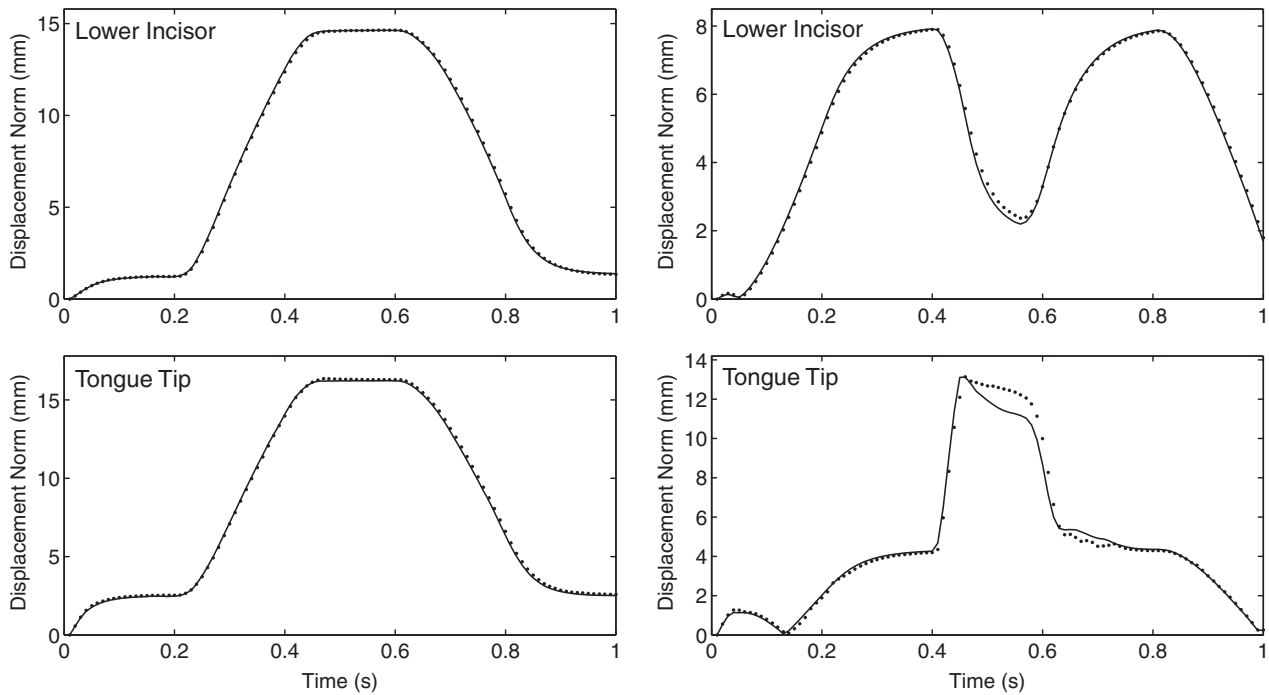
Figure 17. Comparison of output displacements computed with 1 ms (solid lines) and 5 ms (dotted lines) integration steps for the tongue tip (top) and lower incisor (bottom) during jaw opening (left) and tongue–palate contact (right) simulations.

There are a number of ways we believe these solution times can be further improved. Some preliminary experiments with iterative solvers (including ones built into Pardiso) suggest possible speed improvements of 5 to 10 times, although how best to employ iterative techniques with unilateral constraints is not completely clear. Iterative techniques are also amenable to implementation on GPUs. Alternatively, Pardiso itself can be run on multicore technology.

5.4. Stability and contact handling

Most of the simulation tasks described above could be computed stably using a time step between 5 and 10 ms. It should be possible to increase this step size using either an adaptive or fully implicit integration scheme, although the latter will increase the computational cost per step.

When stability problems occur, they are often related to incompressibility, which is usually enforced using bilateral constraints to restrict the divergence of the FEM elements. Incompressibility will occasionally also cause a small amount of hourglassing [6]. Both problems can sometimes be avoided by instead using the B-bar method [37], in which the amount of incompressibility is controlled through the material's bulk modulus parameter [7]. This is supported in ArtiSynth.

There are also some areas in which the contact handling can be improved. First, as indicated in Section 3.6, it would be useful to provide edge–edge-type contacts to better handle situations with low-resolution elements. Second, contact stability can occasionally be a problem: correcting a contact's interpenetration can cause it to separate so that it is not detected as a contact on the subsequent time step, even if the relevant surfaces are being pressed together. The result can be a small amount of jitter as the surfaces continuously interpenetrate and separate. ArtiSynth controls this by limiting the position correction so as to maintain a small amount of interpenetration, but this approach is somewhat *ad hoc*.

The contact jitter problem is a part of the larger issue of contact-coherence, or minimizing the change in contact constraints between time steps. Improved contact-coherence will not only reduce jitter, but also the extent to which the constraint configuration in (6) changes between steps, in turn enabling faster solutions.

6. CONCLUSION

We believe that the simulation framework provided by the ArtiSynth platform is a promising approach to furthering the state-of-the-art for dynamic biomechanical models with combined soft and rigid structures. Our system is intended to bridge the capabilities provided by traditional FEM-based techniques on the one hand and multibody techniques on the other. We have shown that the dynamic simulation of such systems is complicated by the need to combine the velocity solve, which includes a non-diagonal $\hat{\mathbf{M}}$, with both bilateral and unilateral constraints, resulting in a large sparse mixed LCP.

The size of this LCP can be reduced by removing the degrees-of-freedom associated with interbody attachments. Collisions involving deformable bodies can be handled by using the inter-section contour between their surface meshes to isolate interpenetrating FEM nodes, which are in turn used to generate unilateral contact constraints. The LCP itself is solved by (1) treating some contact constraints as bilateral for the duration of each step, hence reducing the number of unilateral constraints, (2) solving the linear part of the LCP using a direct solver (Pardiso), and (3) using a Schur complement to reduce the remaining unilateral constraints into a (small) dense LCP. The resulting solver performs considerably faster than many existing commercial FEM packages. Contributing reasons for this may include (a) our use of semi-implicit time solves and (b) handling contact constraints in a step-wise, impulse-based fashion. With further improvements, possibly based on iterative and/or multicore techniques, we are optimistic about achieving completely interactive simulation times for models of the sort we have described.

Our model of coupled jaw–tongue–hyoid dynamics is a significant step toward a complete model of orofacial biomechanics. We simulated a number of tasks, including unilateral chewing and tongue motions used in speech production, to test the model and evaluate the importance of jaw–tongue coupling. The results for tongue velocity and maximum tongue–palate pressure were consistent with the literature data and showed that coupling is indeed important; in particular, the presence of passive tongue tissue between the jaw and hyoid appears to significantly resist jaw movement.

Modeling dynamic interactions of a jaw–tongue–hyoid model required all the features of our system. For example, the soft tissue deformation of the tongue coupled to movements of the jaw and the impact on the hyoid required the tight coupling of FEM and rigid body simulation. The contact of the tongue tip to the palate used the contact detection mechanisms. The joint constraints depended upon the bilateral constraint implementation. As the jaw–tongue–hyoid model dynamics incur high velocities and large forces, maintaining a stable solve was critical. Further, the high dimensionality of muscle control in the model requires fast simulation times to enable effective exploration and experimentation. The ArtiSynth system embeds all these necessary functions in an interactive simulation environment. Thus, ArtiSynth is an effective simulation environment for problems that require coupled soft-tissue/hard-tissue biomechanics that include bilateral and unilateral constraints.

REFERENCES

1. Perkell JS. Physiology of speech production: results and implications of a quantitative cineradiographic study. *Research Monograph No. 53*. MIT Press: Cambridge, MA, 1969.
2. Huang Y, White D, Malhotra A. Use of computational modeling to predict responses to upper airway surgery in obstructive sleep apnea. *Laryngoscope* 2007; **117**:648–653.
3. Lundy D, Smith C, Colangelo L, Sullivan P, Logemann J, Lazarus C, Newman L, Murry T, Lombard L, Gaziano J. Aspiration: cause and implications. *Otolaryngology—Head and Neck Surgery* 1999; **120**(4):474–478.

4. Buchaillard S, Brix M, Perrier P, Payan Y. Simulations of the consequences of tongue surgery on tongue mobility: implications for speech production in post-surgery conditions. *International Journal of Medical Robotics and Computer Assisted Surgery* 2007; **3**(3):252.

5. Chabanas M, Luboz V, Payan Y. Patient specific finite element model of the face soft tissues for computer-assisted maxillofacial surgery. *Medical Image Analysis* 2003; **7**(2):131–151.

6. Belytschko T, Liu WK, Moran B. *Nonlinear Finite Elements for Continua and Structures*. Wiley: New York, 2000.

7. Bonet J, Wood RD. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press: Cambridge, 2008.

8. Shabana AA. *Dynamics of Multibody Systems*. Cambridge University Press: Cambridge, 1998.

9. Buchaillard S, Perrier P, Payan Y. A biomechanical model of cardinal vowel production: muscle activations and the impact of gravity on tongue positioning. *Journal of the Acoustical Society of America* 2009; **126**(4):2033–2051.

10. Perkell JS. A physiologically-oriented model of tongue activity in speech production. *Ph.D. Thesis*, Speech Communication Group, Massachusetts Institute of Technology, 1974.

11. Laboissiere R, Ostry D, Feldman A. The control of multi-muscle systems: human jaw and hyoid movements. *Biological Cybernetics* 1996; **74**(4):373–384.

12. Payan Y, Perrier P. Synthesis of VV sequences with a 2D biomechanical tongue model controlled by the Equilibrium Point Hypothesis. *Speech Communication* 1997; **22**(2–3):185–205.

13. Sanguineti V, Laboissiere R, Ostry DJ. A dynamic biomechanical model for neural control of speech production. *The Journal of the Acoustical Society of America* 1998; **103**:1615–1627.

14. Payan Y, Chabanas M, Pelorson X, Vilain C, Levy P, Luboz V, Perrier P. Biomechanical models to simulate consequences of maxillofacial surgery. *Comptes Rendus Biologies* 2002; **325**(4):407–417.

15. Perrier P, Payan Y, Zandipour M, Perkell JS. Influences of tongue biomechanics on speech movements during the production of velar stop consonants: a modeling study. *The Journal of the Acoustical Society of America* 2003; **114**(3):1582–1599.

16. Sifakis E, Neverov I, Fedkiw R. Automatic determination of facial muscle activations from sparse motion capture marker data. *SIGGRAPH '05*: *ACM SIGGRAPH 2005 Papers*, 2005; 417–425.

17. Nazari M, Perrier P, Chabanas M, Payan Y. Simulation of dynamic orofacial movements using a constitutive law varying with muscle activation. *Computer Methods in Biomechanics and Biomedical Engineering* 2010; **13**(4):469–482.

18. Hannam A, Stavness I, Lloyd J, Fels S. A dynamic model of jaw and hyoid biomechanics during chewing. *Journal of Biomechanics* 2008; **41**(5):1069–1076.

19. Tuijta M, Koolstraa JH, Lobbezoob F, Naeijeb M. Differences in loading of the temporomandibular joint during opening and closing of the jaw. *Journal of Biomechanics* 2010; **43**(6):1048–1054.

20. Dang J, Honda K. Construction and control of a physiological articulatory model. *The Journal of the Acoustical Society of America* 2004; **115**(2):853–870.

21. Gerard J, Perrier P, Payan Y. *3D Biomechanical Tongue Modelling to Study Speech Production*. Psychology Press: New York, U.S.A., 2006; 85–102.

22. Hunter E, Titze I, Alipour F. A three-dimensional model of vocal fold abduction/adduction. *The Journal of the Acoustical Society of America* 2004; **115**:1747.

23. Anitescu M, Potra FA. A time-stepping method for stiff multibody dynamics with contact and friction. *International Journal for Numerical Methods in Engineering* 2002; **55**(7):753–784.

24. Potra FA, Anitescu M, Gavrea B, Trinkle J. A linearly implicit trapezoidal method for integrating stiff multibody dynamics with contact, joints, and friction. *International Journal for Numerical Methods in Engineering* 2006; **66**(7):1079–1124.

25. Lunk C, Simeon B. Solving constrained mechanical systems by the family of Newmark and α-methods. *Journal of Applied Mathematics and Mechanics* (*ZAMM*) 2006; **86**(10):772–784.

26. Shinar T, Schroeder C, Fedkiw R. Two-way coupling of rigid and deformable bodies. *SCA '08*; *Proceedings of the 2008*; *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Dublin, Ireland, 2008; 95–103.

27. Meseure P, Davanne J, Hilde L, Lenoir J, France L, Triquet F, Chaillou C. A physically-based virtual environment dedicated to surgical simulation. *Proceedings of the International Conference on Surgery Simulation and Soft Tissue Modeling*, Juan-Les-Pins, France, 2003; 38–47.

28. Saupin G, Duriez C, Cotin S. *Contact Model for Haptic Medical Simulations*. Lecture Notes in Computer Science, vol. 5104. Springer: Berlin, 2008; 157–165.

29. Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelman E, Thelen DG. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering* 2007; **54**(11):1940–1950.

30. Weiss J, Maas S. FEBio—a finite element tool for biomechanics, 2007. Available from: http://imechanica.org/node/1897.

31. Cavusoglu MC, Goktekin T, Tendick F. Gipsi: a framework for open source/open architecture software development for organ-level surgical simulation. *IEEE Transactions on Information Technology in Biomedicine* 2006; **10**(2):312–322.

32. Montgomery K, Bruyns C, Brown J, Sorkin S, Mazzella F, Thonier G, Tellier A, Lerman B, Menon A. Spring: a general framework for collaborative, real-time surgical simulation. In *Medicine Meets Virtual Reality*

02/10: *Digital Upgrades*, *Applying Moore's Law*, Westwood JD, Hoffman HM, Robb RA, Stredney D (eds). IOS Press: Amsterdam, 2002; 296–303.

33. Allard J, Cotin S, Faure F, Bensoussan PJ, Poyer F, Duriez C, Delingette H, Grisoni L. Sofa an open source framework for medical simulation. In *Medicine Meets Virtual Reality 15—In Vivo*, *In Vitro*, *In Silico*: *Designing the Next in Medicine*, Westwood JD, Haluck RS, Hoffman HM, Mogel GT, Phillips R, Robb RA, Vosburgh KG (eds). IOS Press: Amsterdam, 2007; 13–18.

34. Fels S, Vogt F, van den Doel K, Lloyd J, Guenter O. Artisynth: towards realizing an extensible, portable 3d articulatory speech synthesizer. *International Workshop on Auditory Visual Speech Processing*, Parksville, Canada, 2005; 119–124.

35. Stavness I, Hannam A, Lloyd J, Fels S. Predicting muscle patterns for hemimandibulectomy models. *Computer Methods in Biomechanics and Biomedical Engineering* 2010; **13**(4):483–491.

36. Müller M, Gross M. Interactive virtual materials. *GI '04*: *Proceedings of Graphics Interface*, London, Canada, 2004; 239–246.

37. Hughes TJR. *The Finite Element Method*: *Linear Static and Dynamic Finite Element Analysis*. Dover Publications: New York, 2000.

38. Miller K, Joldes G, Lance D, Wittek A. Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. *Communications in Numerical Methods in Engineering* 2007; **23**(2):121–134.

39. Lacoursière C. Ghosts and machines: regularized variational methods for interactive simulations of multibodies with dry frictional contacts. *Ph.D. Thesis*, Computer Science Department, Umea University, Sweden, 2007.

40. Anitescu M, Hart G. A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contact and friction. *International Journal for Numerical Methods in Engineering* 2004; **60**(14):2335–2371.

41. Richard W, Cottle RES, Pang J-S. *The Linear Complementarity Problem*. Academic Press: New York, 1992.

42. Schenk O, Gärtner K. Solving unsymmetric sparse systems of linear equations with PARDISO. *Future Generation Computer Systems* 2004; **20**(3):475–487.

43. Aftosmis MJ, Berger MJ, Melton J. Robust and efficient cartesian mesh generation for component-based geometry. *AIAA Journal* 1998; **36**(6):952–960.

44. Edelsbrunner H, Mücke EP. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics* 1990; **9**(1):66–104.

45. Guendelman E, Bridson R, Fedkiw R. Nonconvex rigid bodies with stacking. *SIGGRAPH '03*: *ACM SIGGRAPH 2003 Papers 2003*, 2003; 871–878.

46. Brogliato B, Acary V. *Numerical Methods for Nonsmooth Dynamical Systems*. Springer: Berlin, 2008.

47. Chouly F, Van Hirtum A, Lagree P, Paoli J, Pelorson X, Payan Y. *Simulation of the Retroglossal Fluid–Structure Interaction During Obstructive Sleep Apnea*. Lecture Notes in Computer Science, vol. 4072. Springer: Berlin, 2006; 48–57.

48. Luboz V, Chabanas M, Swider P, Payan Y. Orbital and maxillofacial computer aided surgery: patient-specific finite element models to predict surgical outcomes. *Computer Methods in Biomechanics and Biomedical Engineering* 2005; **8**(2):259–265.

49. Hannam AG, Stavness I, Lloyd JE, Fels S, Miller A, Curtis D. A comparison of simulated jaw dynamics in models of segmental mandibular resection versus resection with alloplastic reconstruction. *Journal of Prosthetic Dentistry* 2010; **104**(3):191–198.

50. Perrier P, Ostry D, Laboissière R. The equilibrium point hypothesis and its application to speech motor control. *Journal of Speech and Hearing Research* 1996; **39**(2):365.

51. Feldman A. Once more on the equilibrium-point hypothesis (lambda model) for motor control. *Journal of Motor Behavior* 1986; **18**(1):17.

52. Stavness I, Hannam AG, Lloyd JE, Fels S. *An Integrated, Dynamic Jaw and Laryngeal Model Constructed From CT Data*. Lecture Notes in Computer Science, vol. 4072. Springer: Berlin, 2006; 169–177.

53. Bucki M, Nazari M, Payan Y. Finite element speaker-specific face model generation for the study of speech production. *Computer Methods in Biomechanics and Biomedical Engineering* 2010; **13**(4):459–467.

54. Muto T, Kanazawa M. Positional change of the hyoid bone at maximal mouth opening. *Oral Surgery*, *Oral Medicine and Oral Pathology* 1994; **77**(5):451–455.

55. Lowe A, Johnston W. Tongue and jaw muscle activity in response to mandibular rotations in a sample of normal and anterior open-bite subjects. *American Journal of Orthodontics* 1979; **76**(5):565–576.

56. Schmidt-Nowara W, Lowe A, Wiegand L, Cartwright R, Perez-Guerra F, Menn S *et al*. Oral appliances for the treatment of snoring and obstructive sleep apnea: a review. *Sleep* 1995; **18**(6):501–510.

57. Utanohara Y, Hayashi R, Yoshikawa M, Yoshida M, Tsuga K, Akagawa Y. Standard values of maximum tongue pressure taken using newly developed disposable tongue pressure measurement device. *Dysphagia* 2008; **23**(3):286–290.

58. Badin P, Serrurier A. Three dimensional modelling of speech organs: articulatory data and models. *IEICE Technical Report*, vol. 106(177), 2006; 29–34.

59. Hiiemae K, Palmer J. Tongue movements in feeding and speech. *Crtical Reviews in Oral Biology and Medicine* 2003; **14**(6):413–429.

60. Ono T, Hori K, Tamine K, Maeda Y. Evaluation of tongue motor biomechanics during swallowing—from oral feeding models to quantitative sensing methods. *Japanese Dental Science Review* 2009; **45**(2):65–74.

61. Weiss J, Maker B, Govindjee S. Finite element implementation of incompressible, transversely isotropic hyperelasticity. *Computer Methods in Applied Mechanics and Engineering* 1996; **135**(1–2):107–128.