Real-Time Simulation and Interaction of Percussion Gestures with Sound Synthesis

Alexandre Bouënard * † Sylvie Gibet * Marcelo M. Wanderley †

(*) SAMSARA / VALORIA, Université de Bretagne Sud, Vannes, France

(†) IDMIL / CIRMMT, McGill University, Montreal, Qc., Canada

Abstract

Virtual characters playing virtual musical instruments must interact in real-time with the sounding environment. Dynamic simulation is a promising approach to finely represent and modulate this interaction. Moreover, captured human motion can provide a database covering a large variety of gestures with various levels of expressivity. We propose in this paper a physics-based environment in which a virtual percussionist is dynamically controlled and interacts with a physics-based sound synthesis algorithm. We show how an asynchronous architecture, including motion and sound simulation, as well as visual and sound outputs can take advantage of the parameterization of both gesture and sound that influence the resulting virtual instrumental performance.

Keywords: Motion Control, Physics Based Computer Animation, Sound Synthesis

Introduction

Playing a musical instrument involves complex human behaviours. While performing, a skilled musician is able to precisely control his motion and to perceive both the reaction of the instrument to his actions and the resulting sound. This relationship between performer actions and the effects on the produced sound is crucial for understanding the mechanisms underlying musical learning or performance. Transposing these real-world experiences into virtual environments provides the possibility to explore novel solutions for designing virtual characters interacting with virtual musical instruments, and by extension for designing animated entities interacting with objects producing realistic sounds.

This paper proposes a physically based framework, in which a virtual character dynamically interacts with a physical simulated percussive instrument. Our goal is to show the influence of the motion control on the produced sound. This influence may be characterized by the subtle physical interactions that occur as the stick makes contact with the drum membrane, as well as the characteristics of the preparatory gesture. In order to take into account the main characteristics of real performers, our approach combines human motion data and physically based simulation with the goal of achieving compelling percussion gestures and producing convincing contact information.

This approach provides a way to interactively manipulate and process both gesture and sound data. Since these data have a physical meaning, their manipulation should guarantee the preservation of the realism of the generated motion and sound. Furthermore, this multisensory virtual environment opens new perspectives to explore gesture and sound parameters, as well as the mapping between them. It also enhances the perception (gestural, visual and auditory) of such complex experiences.

In this paper, we discuss the interaction and mapping between physics-based synthesis of motion and sound, especially with the focus on percussion motion. We first review related literature and compare our approach to other existing techniques. Next we present an overview of our integrated system. We present details on the physics-based motion control approach and the way it exploits motion capture data, and then propose an interaction scheme between motion and sound synthesis. The main results are then presented and discussed. Finally, we conclude with further perspectives.

Related Work

Three main problems have to be addressed in animating characters performing percussion gestures. First, physically based simulation driven by motion capture data has to be considered, so that the physical interaction with the virtual world can be taken into account, and the generated movements can be driven by real examples. A second critical issue is the sound synthesis techniques which have to be used so that an effective mapping can be implemented between parameters generated from motion and sound synthesis parameters. Finally, the synchronization of different modalities and its use in interactive applications needs to be considered.

Controlling adaptative and responsive virtual characters has been intensively investigated in computer animation research. Most of the contributions have addressed the control of articulated figures using robotics-inspired proportionnal derivative (PD) controllers [1]. This has inspired many works for handling different types of motor tasks such as walking, running [2], composing these tasks[3] and easying the hard and time-consuming process of tuning such PD controllers [4]. More related to our work are hybrid methods combining physics-based controllers and kinematic motion data, which aim at associating the advantage of user controllability of kinematics methods and the responsiveness of dynamics controllers. Our work is based on the tracking of motion capture data but differs from previous works by having a fully dynamically controlled character. The specificity of our contribution lies also in the integration and the possible collaboration between inverse kinematics (IK) and inverse dynamics (ID) controllers, rather than handling strategies for transtionning between kinematic and dynamic controllers [5, 6, 7]. We can also find in [8] the use of IK as a pre-process for modifying the original captured motion and simulating it on a different character anthropometry, we rather use IK as a basis of our hybrid control method for specifying the control of a dynamic character from end-effector trajectories. This hybrid collaboration is particularly consistent for the synthesis of such ballistic motion that is percussion performance, which is not take into account in related contributions [8, 9].

Alongside to the dynamic animation of virtual characters, physics-based sound synthesis methods have been widely studied, with the analogous goal of creating adaptative sounds towards changes in environment or objects properties. Since the introduction into the graphics community of modeling the surface vibrations of objects for sound synthesis [10], most of works have either focused on the direct use of vibrational modes [11] based on measurements [12], or the simulation of surface vibrations using the finite element method [13]. More recently, the high interest on modal synthesis has led to acceleration algorithms for handling large and complex environments [14]. Our work involves in a similar manner physical models based on modal synthesis, but we point out to the difficulty of relating these sound synthesis schemes to their cause in the context of music performance, i.e. the instrumental gesture of a virtual performer. Here the focus is on the simulation of instrumental (percussion) gestures, as well as on the mapping between motion and sound synthesis.

We therefore propose an architecture for physically exploring the interaction between motion and sound synthesis that eases the synchronization of different modalities and heterogenous data types (motion capture, movement simulation, sound control parameters). Previous works also involved a motion-driven approach for the synchronized generation of soundtracks from animations [15]. But as recalled in [16], despite many recent improvements, most of computer animation and simulation frameworks are still reluctant to integrate such synchronization method, moving therefore away from the interactive realism and presence that sound can give to computer animation applications. Our goal is to show how a physics modeling of the motion-sound interaction, combined with a carefully-designed system architecture can be of interest to that mean. Such a contribution has been proposed for haptic rendering systems [17, 18], but to our knowledge it has not yet been exploited for the simulation and interaction of instrumental gestures with physics-based sound synthesis.

Overview

The architecture of our system is presented in Figure 1, allowing the physics simulation of percussion gestures to interact with sound synthesis processes. The physics-based motion control of the virtual character involves a motion capture database for planning percussion gestures, as well as the hybrid combination of IK and ID controllers. The interaction expresses the mapping between the percussion motion simulation and the sound synthesis module, and features a specially designed architecture for accelerating and easying this process.

Physics-based Motion Control

The approach that we adopt for physically controlling percussion gestures from motion capture data is described in Figure 2. It involves the physical modeling of the virtual percussionist from pre-recorded percussion performances, and the controller which tracks motion capture data, requiring either a simple ID control mode, or a combined IK and ID control mode.

Virtual Character Modeling

The physics-based modeling of the virtual character is composed of rigid bodies articulated by mechanical joints. Motion capture data of percussion performances are used to physically model and parameterize both the anthropometry and the mechanical joints of the virtual character, making a direct correspondence between the real performer and the virtual character. The physical properties of each rigid body composing the virtual character, such as mass, size of the limbs, density and inertia matrix are consistent with the anthropometry extracted from motion capture data.

In addition, each mechanical joint has three rotational degrees of freedom, restricted to the angular limits of the human body, in order to avoid non realistic motion. Two formulations are available to extract the ”lower” and ”upper” limits of the angular joints, based on a statistical analysis of the motion capture data. The first formulation uses the Euler angles representation, and computes basic statistical features to characterize joint limits. The singularity of this represention is however frequent (gimbal lock), therefore we propose another formulation based on the quaternionic representation. Following the approach of [19], we compute joint limits in the quaternion space. The rotational (quaternionic) trajectory of a joint over time is transposed in the tangential space of its quaternionic mean, in which an SVD decomposition is applied. The resulting eigen values and axes are used for representing the motion distribution and for computing the joint limits (Algorithm 1).

Motion Control

Our approach to dynamic character control uses percussion gestures from a motion capture database, allowing to take into account all the variability and expressiveness of real percussion gestures. The variability can be due to various percussion performances using different drumstick grips, various beat impact locations and several musical playing variations. We propose two ways for achieving the motion control (Figure 2), either by tracking motion capture data in the joint space (angular trajectories), or tracking end-effector trajectories in the 3D cartesian space. Tracking motion capture data in joint space requires ID control, whereas tracking in the end-effector space requires both IK and ID control. In this latter case, the two inversion processes are strongly linked.

Tracking Motion Capture Data in Joint Space

This control mode is related to motion capture data tracking by using ID controllers. Angular trajectories (Θ^T) are first extracted from motion capture data, and used to drive the fully dynamically controlled virtual character. We use traditional proportional-derivative feedback controllers, modeled as damped springs and parameterized by manually-tuned damping and stiffness coefficients (k_d, k_s). Knowing the current state of the mechanical joint (Θ^S, $\dot{\Theta}^S$) and the joint target (Θ^T) to be reached, the torque (τ) is computed and exerted on the articulated rigid bodies, accordingly to equation (1).

$$\tau = k_s.(\Theta^S - \Theta^T) - k_d.\dot{\Theta}^S \quad (1)$$

Tracking Motion Capture Data in 3D Cartesian Space

A more intuitive physics control of the virtual character is proposed by combining IK and ID controllers. Instead of directly tracking angular trajectories from the motion capture database, this tracking mode consists in extracting end-effector positions in the 3D Cartesian space. From these Cartesian targets, an IK method computes (equation 2) the kinematic postures (joint vector $\Theta^T = \{\Theta_1, ..., \Theta_n\}$), which are used as the desired input of the ID controllers (Algorithm 2), thus providing the required torques to control the physical character (Figure 2).

$$\Delta\Theta^T = -\lambda . J^+_\Theta . (X^S - X^T), \;\; \Theta^T = \Theta^S + \Delta\Theta^T \tag{2}$$

J^+_Θ is the Jacobian pseudo-inverse of the system to be controlled, X^S and X^T represent the current and target end-effector positions in the Cartesian space. In this paper, we implemented a slightly modified version of the pseudo-inverse that guaranties that singularities are avoided, referred to as the damped or singularity robust (SR) pseudo-inverse [20]. One may equally use other IK techniques, such as the transpose of the Jacobian J^t, or the learning technique described in [21].

Using such an IK formulation necessitates the computation of the Jacobian matrix, and therefore we defined an equivalent representation of the articulated chain, both in the kinematics and dynamics spaces. The main difficulty with the coupling of both kinematics and dynamics controllers is that the convergence of the IK algorithm is added to the difficulty of tuning the parameters of the PD dynamic controllers. But this approach enables the manipulation of motion capture data in the 3D Cartesian space (configuration X^T) instead of the angular space (Θ^T), which is more consistent and intuitive for controlling percussion gestures, by using end-effector trajectories, for instance drumsticks extremities.

Interaction between Motion and Sound Synthesis

We propose a general architecture (Figure 3) that allows for simultaneously and asynchronously running the physics simulation of percussion gestures, graphics and sound rendering processes, as well as handling their interaction in real-time. This architecture is effective for managing different modalities and data types, and for specifying at the physics level the mapping between these.

Asynchronous Client-Server Architecture

Our system allows the multimodal integration, interaction and synchronization of visual and sounding media, and is composed of four components represented as physics, graphics, interaction and sound managers.

Multimodal processes such as graphics and sound rendering are fundamently different, and achieving a real-time interaction between the two can be hazardous. The first difficulty that appears when managing such media is their discrepency in time constants: graphics rendering is usely admitted effective at about 33Hz, whereas sound rendering needs a higher time sample ate around 44kHz. Moreover, the graphics rendering is the visible layer of a far more demanding process that is the physics simulation, requiring the handling of two other different time rates since our method relies on motion capture tracking: the original motion capture time rate and the time step of the physics simulation. An approach could consist in running synchronously every manager at the sound rate, but such an approach would fall short in real-time considerations.

We propose an asynchronous client-server scheme for handling the interaction between the physics, graphics and sound managers. This enables the possible distribution of the different managers on distinct platforms, thus reducing the computational cost of each manager and its impact on the others. For this purpose, we adopt the Open Sound Control (OSC) communication protocol [22] which is traditionally used in the computer music community and allows the real-time exchange of multimedia data flows. The asynchronous exchange of data is materialized here by the mapping between the output of motion synthesis and the input of sound synthesis. Events produced during the physical simulation are dealt by the interaction module which feeds the sound synthesis processes and the visual outputs.

Motion-Sound Physics Mapping

For sound synthesis, we consider in this paper the modal synthesis technique, which is an efficient way of representing the vibrations of resonating objects as the motion simulation of systems composed of masses connected with springs and dampers. This physically-based approach enables the direct modeling of the contact force impact. According to [23], this force depends on the state (displacement and velocity) of the colliding modal objects.

The interaction manager includes a collision detection algorithm that can retrieve the physical features of any contact event produced when the drum is excited. In particular it provides information on the impact position, velocity and force (direction and amplitude), which can be the input to the physics-based sound synthesis process, in a direct one-to-one mapping.

As for the sound synthesis system, it allows for the real-time parameterization of the membrane properties (size, mass, tension), as well as the parameters of the modal synthesis (number of modes, resonances), thus rendering varied sound feedback effects .

Results

We first present results regarding the physics-based control scheme described in the paper, where original motion capture data are compared both to the data synthesized with the ID controller (Joint Space Tracking) and the combination of IK and ID controllers (Cartesian Space Tracking). Secondly, an application of the proposed architecture, namely the integration of the OSC protocol and the development of a user interface in which the simulation of percussion performances can be parameterized.

Physics Motion Capture Tracking: Joint Space vs. Cartesian Space

The physical model of the virtual character is composed of 19 joints, totalling 57 degrees of freedom. The Open Dynamic Engine [24] is used for the overall motion simulation. Masses, inertia, limb lengths, as well as joint limits are estimated from real percussion performers. The results obtained by the two tracking modes, used to physically control the virtual percussionist are compared: a) motion capture tracking in joint space, involving an ID controller, and b) motion capture hybrid tracking in cartesian space, involving a combination of IK and ID controllers. Concerning the hybrid control mode, different inverse kinematics methods may be combined to ID controllers. In this paper, we implemented the Damped Least Squares method [20], a simple yet more robust adaptation of the pseudo-inverse regarding the singularity of the inverse kinematics problem.

Figure 4 compares raw data from captured motion with the two modes of control (ID only, and the hybrid combination of IK and ID). In these two control modes, we kept the same parameterization of the damped springs of the virtual character. We ran the simulation on a set of pre-recorded percussion gestures (French grip, legato), that were recorded at a sample rate of 250 Hz (we found it high enough to capture the whole body of the performer as well as the drumsticks). The hybrid control scheme tracks one percussion gesture for synthesizing whole arm movements only from the specification of the tip of the drumsticks trajectories. Figure 4 (top) presents the comparison between raw motion capture data and data generated by the IK process. It shows that data generated by the IK formulation are consistent with real ones, especially for the elbow flexion angle that is one of the most important degree of freedom of the arm in percussion gestures (especially during preparatory phases [25]). We finally present the comparison of the two control modes (ID control only and hybrid control) in Figure 4 (bottom). One interesting issue is the accuracy of the hybrid control mode compared to the simple ID control. This observation lies in the fact that the convergence of motion capture tracking is processed in the joint space in the case of ID control, adding and amplifying multiple errors on the different joints and leading to a greater error than processing the convergence in the Cartesian space for the hybrid control. The main drawback of this improvement is however the additional computationnal cost of the IK algorithm which is processed at every simulation step. It provides nevertheless a more flexible motion edition technique for controlling a fully physics-based virtual character, that eases the co-articulation between successive motion units. This is illustrated in Figure 5, which shows the creation of a gesture score from a musical score, and the animation of the virtual percusionist following this score.

Fast-Easy Interaction for Sounding Virtual Character Animation

Our software architecture implements the OSC protocol, without any assumption about the hosting of OSC clients and servers, making it possible to run the graphics, physics and sound managers on distinct computers. This architecture has been successfully tested by running the graphics/physics and sound cores on two different platforms linked by an ethernet connexion, providing an effective and reliable communication between the two.

The user interface presented in Figure 6 was implemented using Pure Data [26], which is considered as the sound OSC server in our architecture. It shows how users can instantiate and access OSC components (top control panels) by modifying, creating and registering new interaction messages between the physics (left control panel), graphics (middle control panel) and sound (right control panel) managers. Users can select different percussion grips (French or German), movement nuances (legato, tenuto, accent, vertical accent and staccato), and different tempi, to be simulated by the virtual percussionist. The parameterization of the visual and sound feedback is also possible. During the simulation, the interface provides users with different sound synthesis models such as the simple playback of sound clips, signal-based sound synthesis, or physics-based sound (modal) synthesis. Every sound synthesis technique proposed by the interface can be tuned in real time. For instance for the modal sound synthesis module, one can tune the parameterization of the drum membrane physics properties (radius size, mass, tension).

Conclusion

We proposed in this paper a physically-enabled environment in which a virtual percussionist can be physically controlled and interact with physics-based sound synthesis schemes. The physics-based control from real percussion performances guarantees to maintain the main characteristics of human motion data while keeping the physical coherence of the interaction with the simulated instrument. Furthermore, the hybrid control mode combining IK and ID controllers leads to a more intuitive way of editing the motion to be simulated only from drumstick extremity trajectories. Alongside, the proposed asynchronous client-server architecture takes advantage of motion and sound physics formulations, generating in real-time virtual percussion performances that can be parameterized from the motion to the sound.

Among the perspectives of such work is the improvement of the interaction between motion simulation and sound synthesis. This can involve the design of a more powerful collision detection module. If we consider the physical simulation of the instrument, it might be interesting to include mechanical interactions, such as hammered or plucked interactions, or even modeling the mechanical structure of a finger or a hand. We also plan to extend the mapping possibilities beween gesture parameters and sound input parameters, and to develop efficient methods for editing percussion captured motion in the 3D Cartesian space.

References

[1] M. Raibert. *Legged Robots that Balance*. MIT Press, 1986.

[2] J. Hodgins, W. Wooten, D. Brogan, and J. O'Brien. Animating Human Athletics. In *Proc. of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 71–78, 1995.

[3] P. Faloutsos, M. van de Panne, and D. Terzopoulos. Composable Controllers for Physics-Based Character Animation. In *Proc. of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 251–260, 2001.

[4] B. Allen, D. Chu, A. Shapiro, and P. Faloutsos. On the Beat!: Timing and Tension for Dynamic Characters. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 239–247, 2007.

[5] V. Zordan and J. Hodgins. Motion Capture-Driven Simulations that Hit and React. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 89–96, 2002.

[6] A. Shapiro, F. Pighin, and P. Faloutsos. Hybrid Control For Interactive Character Animation. In *Proc. of the Pacific Conference on Computer Graphics and Applications*, pages 455–461, 2003.

[7] V. Zordan, A. Majkowska, B. Chiu, and M. Fast. Dynamic Response for Motion Capture Animation. *ACM Transactions on Graphics*, 24(3):697–701, 2005.

[8] V. Zordan and J. Hodgins. Tracking and Modifying Upper-body Human Motion Data with Dynamic Simulation. In *Proc. of Computer Animation and Simulation*, pages 13–22, 1999.

[9] A. Bouënard, S. Gibet, and M. M. Wanderley. Enhancing the Visualization of Percussion Gestures by Virtual Character Animation. In *Proc. of the International Conference on New Interfaces for Musical Expression*, pages 38–43, 2008.

[10] K. van den Doel and D. Pai. Synthesis of Shape Dependent Sounds with Physical Modeling. In *Proc. of the International Conference on Auditory Display*, 1996.

[11] J. O'Brien, C. Shen, and C. Gatchalian. Synthetizing Sounds from Rigid Body Simulations. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 175–181, 2002.

[12] R. Corbett, K. van den Doel, J. Lloyd, and W. Heidrich. Timbrefields: 3d Interactive Sound Models for Real-Time Audio. *Presence: Teleoperators and Virtual Environments*, 16(6):643–654, 2007.

[13] J. O'Brien, P. Cook, and G. Essl. Synthesizing Sounds from Physically Based Motion. In *Proc. of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 529–536, 2001.

[14] N. Bonneel, G. Drettakis, N. Tsingos, I. Viaud-Delmon, and D. James. Fast Modal Sounds with Scalable Frequency-Domain Synthesis. *ACM Transactions on Graphics*, 27(3):1–9, 2008.

[15] T. Takala and J. Hahn. Sound Rendering. *SIGGRAPH Computer Graphics*, 26(2):211–220, 1992.

[16] K. van den Doel, P. Kry, and D. Pai. FoleyAutomatic: Physically-Based Sound Effects for Interactive Simulation and Animation. In *Proc. of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 537–544, 2001.

[17] F. Avanzini and P. Crosato. Integrating Physically based Sound Models in a Multimodal Rendering Architecture. *Computer Animation and Virtual Worlds*, 17(3-4):411–419, 2006.

[18] S. Sinclair and M. M. Wanderley. A Run-time Programmable Simulator to Enable Multimodal Interaction with Rigid Body Systems. *Interacting with Computers*, In press.

[19] M. Johnson. *Exploiting Quaternions to Support Expressive Interactive Character Motion*. PhD thesis, Massachusetts Institute of Technology, 2003.

[20] C. Wampler. Manipulator Inverse Kinematic Solutions based on Vector Formulations and Damped Least Squares. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1):93–101, 1986.

[21] S. Gibet and P. F. Marteau. Expressive Gesture Animation based on Non-Parametric Learning of Sensory-Motor Models. In *Proc. of the International Conference on Computer Animation and Social Agents*, pages 79–85, 2003.

[22] M. Wright. Open Sound Control: an Enabling Technology for Musical Networking. *Organised Sound*, 10(3):193–200, 2005.

[23] F. Avanzini and D. Rochesso. Controlling Material Properties in Physical Models and Sounding Objects. In *Proc. of the International Computer Music Conference*, pages 91–94, 2001.

[24] Open Dynamics Engine. *http://opende.sourceforge.net*, 2009.

[25] A. Bouënard, M. M. Wanderley, and S. Gibet. Analysis of Percussion Grip for Physically Based Character Animation. In *Proc. of the International Conference on Enactive Interfaces*, pages 22–27, 2008.

[26] M. Puckette. Pure Data. In *Proc. of the International Computer Music Conference*, pages 269–272, 1996.

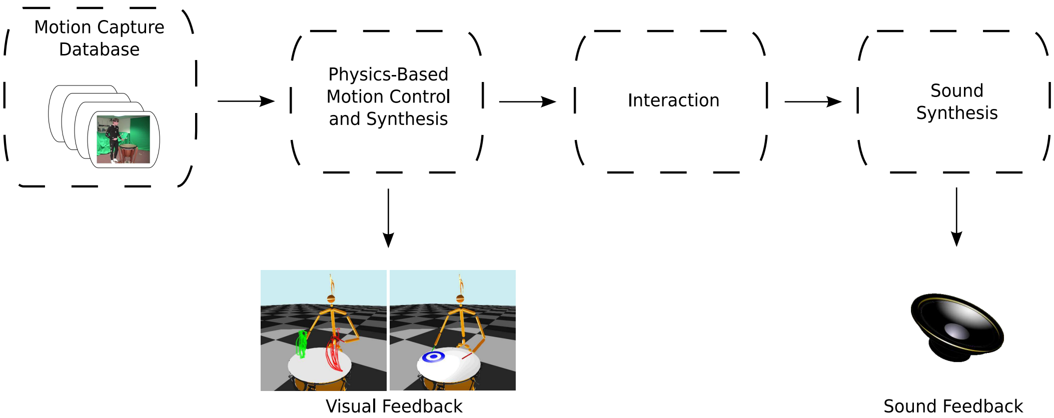

Figure 1: System architecture and multimodal outputs. The *Physics-based Motion Control and Synthesis* step involves a *Motion Capture Database* and results in the *Visual Feedback*. The *Interaction* expresses the mapping between the percussion motion simulation and the *Sound Synthesis*, which results in the *Sound Feedback*.

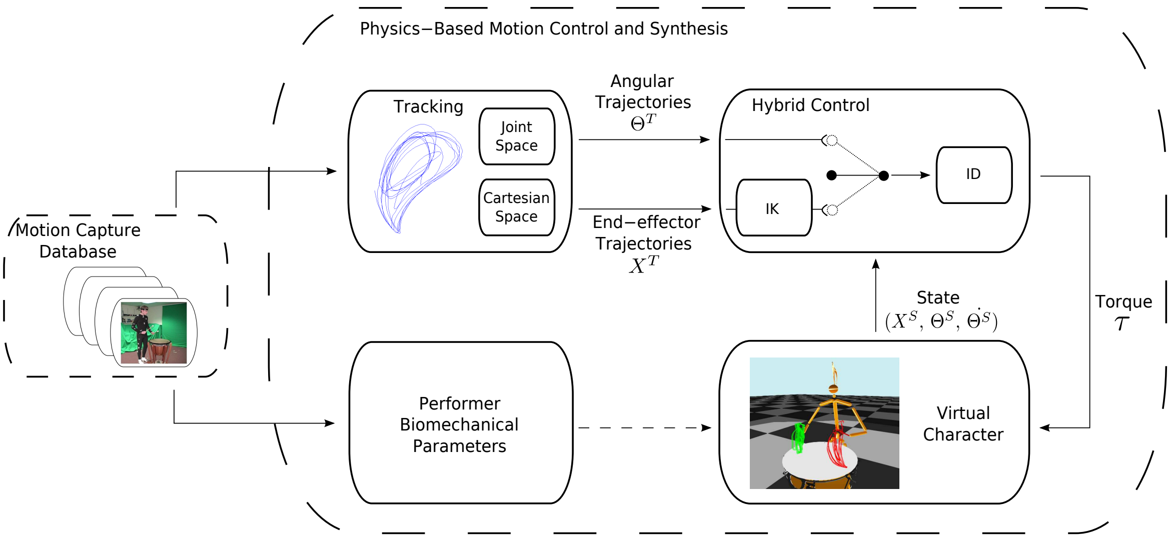

Figure 2: Physics-based Motion Control and Synthesis. The *Motion Capture Database* is used for the physics modeling of the *Virtual Character*, and for expressing two levels of *Tracking*. These two levels allow the physics-based motion capture tracking, either in the *Joint Space* from angular trajectories Θ^T, or in the *Cartesian Space* from end-effector trajectories X^T. The hybrid control involves the combination of inverse kinematics (*IK*) and inverse dynamics controllers (*ID*).

Algorithm 1 Formulating joint limits in the quaternion space

Require: Joint motion expressed as a quaternionic time serie: $Q = \{q_i, i \in [[1 \ldots n]]\}$

Ensure: Joint motion eigen values and axes: $\{\vec{\lambda}, \vec{e}\}$

$\bar{q}$ = mean(Q) $\{\bar{q}$: mean of $Q\}$

for $i = 1$ to n **do**

$q_i^* = \log(\bar{q} * q_i)$ $\{Q^*$: transposition of the motion Q in the tangent space of $\bar{q}\}$

end for

$\{\vec{\lambda}, \vec{e}\} = SVD(Q^*)$ $\{\{\vec{\lambda}, \vec{e}\}$: eigen values and axes$\}$

Algorithm 2 Hybrid control of an articulated chain combining inverse kinematics and inverse dynamics controllers

Require: a dynamic articulated chain: $dAC = \{dS_i, i \in [[1 \ldots n]]\}$,

a target: X^T

Ensure: a set of torques: $\tau = \{\tau_i, i \in [[1 \ldots n]]\}$,

a kinematic configuration: $\Theta^T = \{\Theta_i^T, i \in [[1 \ldots n]]\}$

while simulation **do**

$\{$Inverse Kinematics$\}$

ComputeJacobian(X^T)

$\Theta^T \leftarrow$ InverseKinematics()

$\{$Inverse Dynamics$\}$

for $i = 1$ to n **do**

$\tau_i \leftarrow$ InverseDynamics(dS_i, Θ_i^T)

ApplyTorque(dS_i, τ_i)

end for

Update()

end while

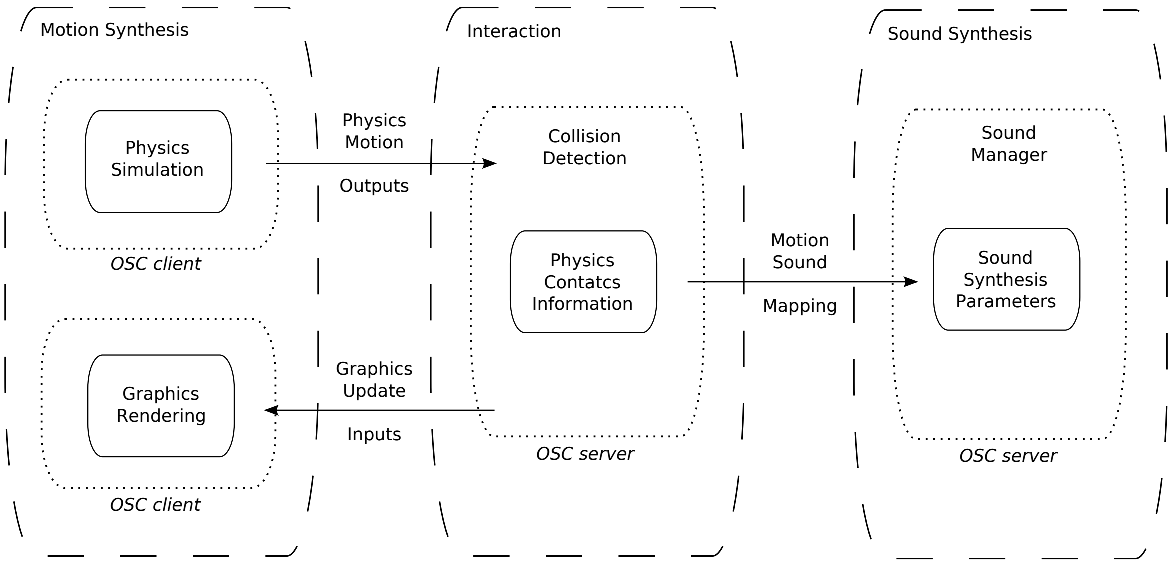

Figure 3: Asynchronous server-client architecture of our system.

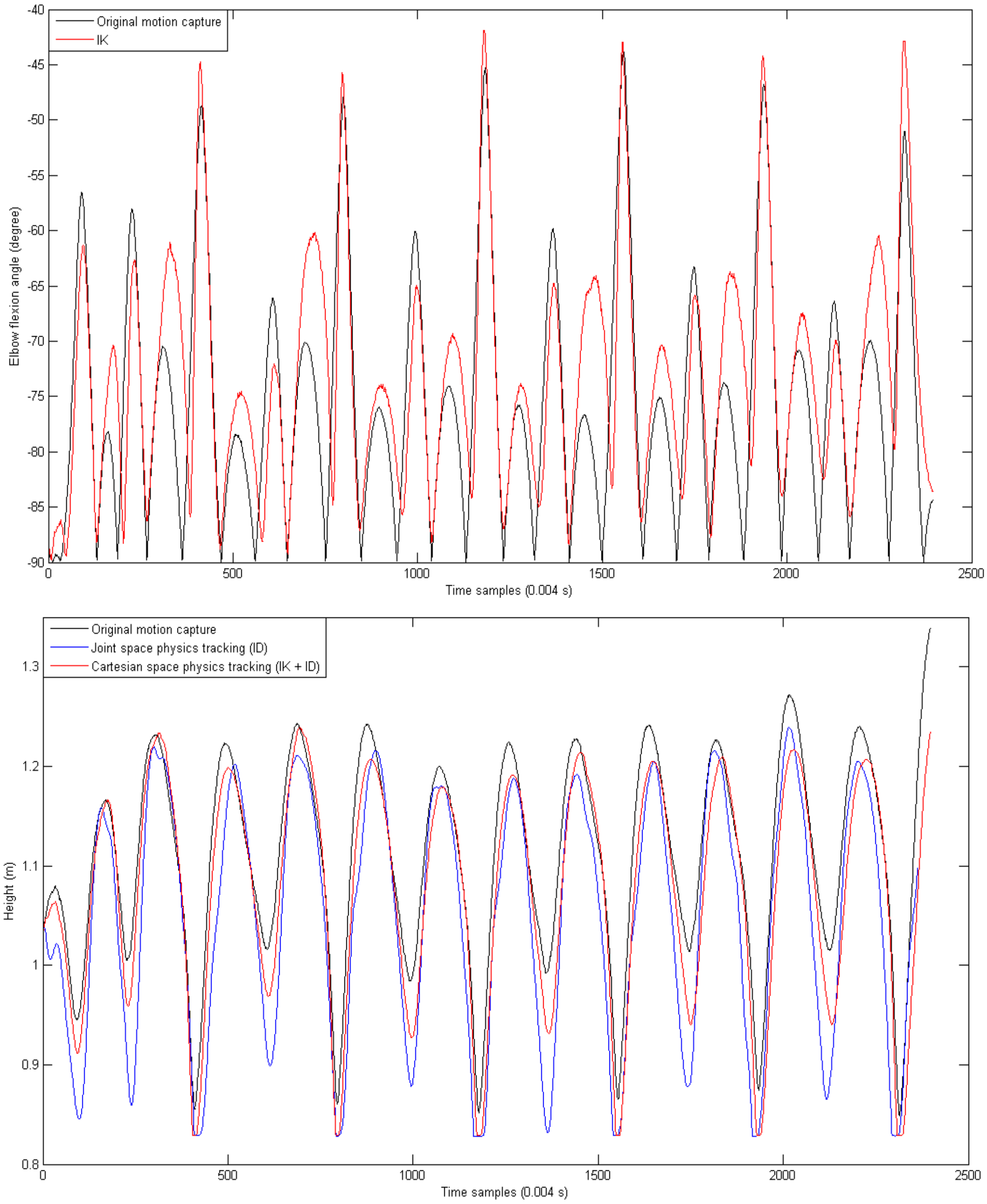

Figure 4: Top: comparison of elbow flexion angle trajectories: original motion data (black) vs. data generated by the IK algorithm (red). Bottom: comparison of drumstick trajectories: original motion capture data (black) vs. joint space (ID) physics tracking (blue) vs. cartesian space (IK + ID) physics tracking (red).

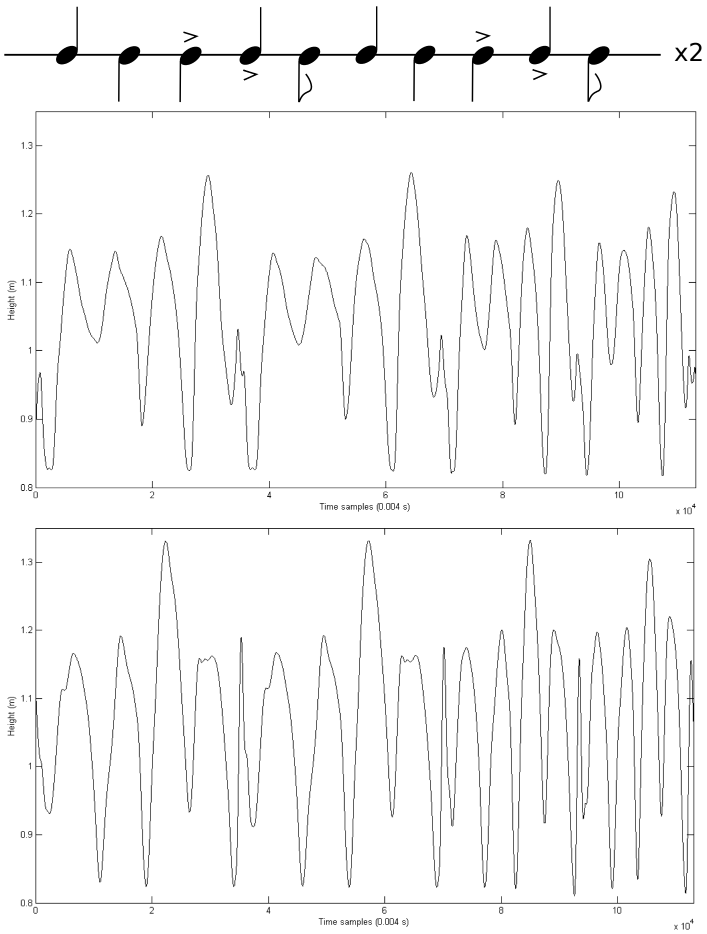

Figure 5: Score (top) performed by the virtual percussionist combining legato and accent beats under tempo variation, and the resulting height of the right (middle) and left (bottom) tip trajectories of the drumsticks during the simulation.

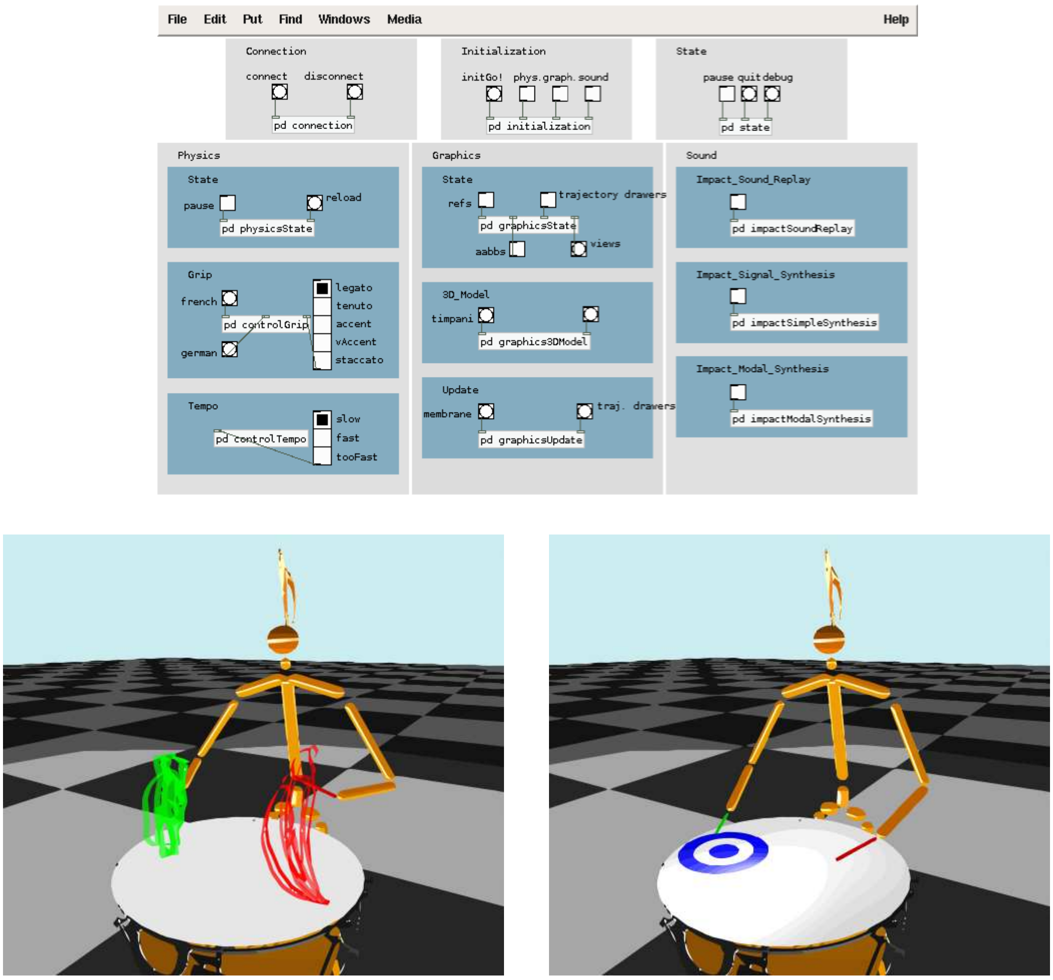

Figure 6: Users can parameterize the percussion gesture to be simulated (drum grip, musical variation, tempo), as well as the graphics rendering and the sound feedback to be used (sound replay, signal-based and physically-based sound synthesis).