FP7-ICT-2007-C
Objective ICT-2007.8.0
FET-Open 222107

NIW
Natural Interactive Walking

Deliverable 4.1

AUDITORY AND HAPTIC AUGMENTATION OF FLOOR SURFACES

Antonio De Sena, Carlo Drioli, Federico Fontana, Stefano Papetti – UNIVR
Rolf Nordahl, Stefania Serafin, Luca Turchet – AAU
Yon Visell, Alvin Law, Jessica Ip, Rishi Rajalingham, Severin Smith, Jeremy Cooperstock – McGill Univ.

Sep. 30, 2009
v. 1.0

Classification: PU

Contents

1 Introduction

The auditory and haptic augmentation of floor surfaces can be used to display a layer of virtual grounds to walkers, by exposing information in the form of ecological cues of floor materials. Such a result can be achieved only if a flexible and efficient synthesis engine is available that, once triggered by an external signal, generates realistic feedback within a few milliseconds. This feedback needs to be informed by the characteristic parameters of the virtual ground, and by the type of event that gives rise to the interaction: only by reproducing the same event on a simulated ground without noticeable latency can this engine synthesize convincing cues to the users.

A software providing all these features is neither available as a commercial product, nor as a research prototype. However, successful designs have been presented in the recent past, inspired by the same idea. Perry Cook's PhOLIEMat, probably the closest prior art, used the Bill's GaitLab synthesis engine to sonify tiles simulating different materials [12]. Furthermore, important related art that was not directly used for sonic and haptic surface material modeling resulted in a number of floor-based interaction systems. Some of them were responsive: for instance, the ADA system [14].

The NIW project aims at improving the current state of the art in responsive floor and shoe systems. This deliverable reports on the results that have been achieved during the first period of the project concerning the auditory and haptic feedback. Such results range from theory to implementation, moving through the realization of synthesis algorithms that can be performed in real time via dedicated environments like puredata (pd) and Max/MSP. The relatively broad scope of this deliverable is motivated by the twofold aspect of this project research, that expects to come up with both an integrated platform in a three-year term, and novel fundamental knowledge that will hopefully pay off in the long term.

To date many of these results are not integrated together yet. Coherently with the project plan, the consortium hopes to come to an acceptable level of integration during the next period, and finally to realize a demostrator exhibiting a tight integration level by the end of the last period. Conversely, the same results form an already rich palette of techniques and models that will enable the consortium to start the experiments scheduled for the next period.

More in detail, the deliverable:

- starts with theoretical achievements obtained in the physical modeling of impact, and in the use of fractal noise in physical models of friction. For its expected long-term impact in the sonic and haptic synthesis of interactive events, physical modeling has been the only selected object of theoretical research;
- proceeds with a presentation of the palette of real-time walking sounds obtained so far. Most of these sounds simulate the interaction with aggregate surfaces such as crumpled and snowy paths, in this reflecting an orientation of the consortium to display virtual grounds that mark a difference with homogeneous physical floors—indeed the use of granular sounds in augmented, responsive objects is not new [35];
- addresses the synthesis of liquid sounds, whose models are traditionally difficult to be simulated accurately and efficiently at the same time, and whose application to walking interfaces is totally unexplored;
- introduces a class of sound synthesis algorithms based on parametric models, that, although researched only for a couple of months, promise to provide accurate walking sounds over different materials at low computational cost;
- cites an experience of limited time made at UNIVR, in which a smartphone was used both to acquire the user's gestures and to display sonic feedback;
- explains the way the synthetic walking sounds are displayed to the users of the interfaces;

- reports on an already existing integrated platform at McGill University, whose multimodal feedback puts the NIW project overall on an already advanced state, and from where much can be learned from the beneficiaries concerning the design of the final demonstrator.

Online material

Some chapters of this deliverable are complemented by multimedia documents that can be downloaded from the NIW website, `www.niwproject.eu`. From the home page, the reader must click on the "Deliverable 4.1" menu item and choose the material from the chapter of interest.

Whenever the content of a paragraph is connected to a multimedia document, the reader is explicitly invited to visit the website.

The software prototypes associated to this deliverable can be compiled and installed in machines that run flext, puredata (also called pd), and Max/MSP:

- flext can be freely downloaded from `http://puredata.info/Members/thomas/flext/`
- puredata is freely available at `http://www.puredata.org`
- Max/MSP can be purchased at `http://www.cycling74.com`.

2 Basic research: Physical sound modeling

Physical modeling is probably the most promising sound synthesis paradigm that can be employed for the production of everyday sounds [45]. By simulating the dynamics of a mathematical description of a system, a physical model computes the sound of this system as part of this simulation. Hence, usually with a minor additional effort it can also generate haptic signals [23].

For such reasons, the consortium is carrying on fundamental research on physical models. While it is not clear whether these results will find application in the short-medium term, particularly before the end of the project, on the other hand they certainly contribute to form a long-term knowledge basis for future application in the sonic and haptic interaction design field.

Specifically, the rest of this chapter reports on the research made during the first period about the modeling of two fundamental ecological events: impact and friction. The former is evolved in an aim to improve its accuracy and stability. The latter is complemented with fractal noise, whose inclusion promises to lead to improved simulations of rubbing events that, for the project purposes, turn to be useful in the design of foot-floor interactions.

2.1 Impact modeling

A non-linear physical model of impact with sound synthesis applications has been reviewed [36], and its properties have been studied using both analytical tools and numerical simulations. Several numerical realizations have been compared, and their shortcomings with regard to the corresponding analytical results have been shown. Special emphasis has been placed on energy consistency. Moreover, it has been shown that by exploiting some analytical results, the inconsistencies of the numerical realizations can be amended, thus restoring the correct energy state of the simulated systems after the impact interaction has ended.

Physical models of impacts between objects are ubiquitous in many areas of science and engineering, including robotics, haptics, computer graphics [33], acoustics [10] and sound synthesis [3]. The energetically-consistent and phenomenologically-plausible behavior of contacting bodies is especially crucial in simulations of interactions based on sustained and repeated impacts, as in rolling, scraping, or bouncing.

With regard to the purposes of the project, the goal of this research is to provide the consortium with a stable and computationally efficient algorithm for the simulation of impact interactions between solids (especially the foot-ground interaction), which could be exploited for both sound synthesis and haptic rendering [2].

Contact sound models

Contact models borrowed from different application fields can serve as a basis for developing models of acoustic phenomena. As an example, in the context of physical sound modeling, the impact model described by Equation (2.2) has already been used to develop an impact sound model [3], namely by substituting the rigid wall component with a generic resonating object. Also, other models of more complex sound phenomena have been based on the very same impact model studied here. As a basic example, a *bouncing* sound model [46] has been obtained by superimposing a constant force, which simulates gravity, on a plain impact sound model. Moreover, a *rolling* sound model [42] has been developed by driving an impact sound model by means of a sophisticated control layer. More precisely, the continuous interaction of a ball rolling on a surface has been modeled as a dense temporal sequence of micro-impacts driven by the geometry of the contacting surfaces and modulated by the ball's asymmetry.

From these application examples it is clear that, even if the errors are generally tolerably small[1] for single impact events, in case of sustained contacts or multiple impacts the energy state of the system can become strongly inconsistent. This is a known issue also in computer graphics, where the constraint of low frame rates makes numerical systems prone to instabilities [33]. An example is that of a steady object in sustained contact with a rigid floor: when the system does not preserve passivity, the object can move upward and bounce. Similar issues are encountered in simulation of haptic contact, where stiffness values are usually limited by requirements on system passivity [11], whereas higher values can cause the system to become unstable, i.e., to oscillate uncontrollably.

2.1.1 Mathematical description

The classic starting point is the Hertz model of collision between two spheres, which can be extended to include internal viscosity [20]. The restitution force in such model is the sum of a nonlinear elastic term – in the form of a power law of compression – and a dissipative component proportional – via a second power law of compression – to the compression velocity. The exponents of the two power laws, as derived for two colliding balls, are 3/2 and 1/2, respectively [25].

The impact model by Hunt and Crossley [24, 28, 15, 26, 19] generalizes the extended Hertz model by considering a variable exponent that accounts for different contact shapes. However, the power laws in the elastic and dissipative term were considered to be equal, thus allowing easier closed-form calculations (in this regard see also Pust and Peterka [40]).

The Hunt-Crossley model is described by the following non-linear equation for the force f:

$$f(x, v) = \begin{cases} kx^\alpha + \lambda x^\alpha v = kx^\alpha \cdot (1 + \mu v) & , \quad x > 0 \\ 0 & , \quad x \leq 0 \end{cases} \tag{2.1}$$

where x is the *compression*, $v = \dot{x}$ is the *compression velocity*, $\alpha > 1$ is the *exponent* of a power law and represents the local shape of contact surfaces, k is the *stiffness coefficient*, $0 \leq \lambda \leq k$ is the *damping coefficient*, and $\mu = \lambda/k$ is a mathematically convenient term. The model above represents a non-linear spring of constant k in parallel with a non-linear damper of constant λ. Indeed, kx^α represents the elastic component, while $\lambda x^\alpha v$ represents the dissipation due to internal friction.

Marhefka and Orin [28] made use of the Hunt-Crossley model in order to represent the impact between a lumped point-mass and a rigid wall (representing a comparatively massive surface which does not move during collision), therefore considering the system described by the equation

$$ma(t) = -f(x(t), v(t)), \tag{2.2}$$

where m is the mass, and a is the mass acceleration. In this very basic case, while the impact interaction lasts, the compression and the compression velocity are respectively equivalent to the displacement and the velocity of the point-mass.

Properties and analytical results

Thanks to the simple form of Equation (2.2), the model can be treated analytically and some of its properties can be inferred. Hereafter we consider as initial conditions $x(0) = 0$ and $\dot{x}(0) = v_{\text{in}}$, that is to say that the point-mass hits the rigid wall with velocity v_{in} at time $t = 0$.

Displacement / Compression It is shown in [28] that from Equation (2.2) follows

$$x(v) = \left[\frac{m(\alpha + 1)}{k\mu^2} \cdot \left(-\mu(v - v_{\text{in}}) + \log \left| \frac{1 + \mu v}{1 + \mu v_{\text{in}}} \right| \right) \right]^{\frac{1}{\alpha+1}} \tag{2.3}$$

[1] Especially if $\tau > 4$ samples, and when using implicit or high-order numerical methods.

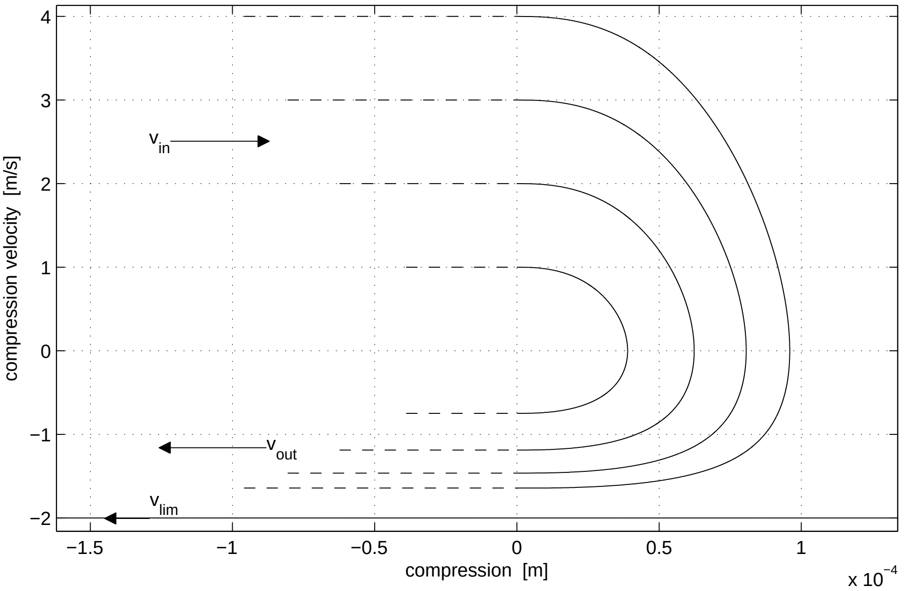


Figure 2.1: Phase portrait. The values of parameters are: $m = 10^{-2}$ kg, $k = 10^9$ N/m$^\alpha$, $\mu = 0.5$ s/m, $\alpha = 1.5$, $v_{\text{in}} = 1 \dots 4$ m/s. Solid lines represent the mass trajectory during contact, dashed lines represent free mass motion.

which can be exploited for plotting the phase portrait on the (x, v) plane of figure 2.1. As figure 2.1 shows, due to viscous dissipation (represented by λ or μ), the velocity after collision v_{out} is always smaller in magnitude than the corresponding v_{in}. Moreover, for increasing v_{in}, v_{out} converges to the limit value v_{lim}. The line $v = v_{\text{lim}}$ represents the trajectory where the elastic and dissipative terms cancel, and separates two regions of the phase space, each of which is never entered by trajectories started in the other.

It can be noted that Equation (2.3) allows to infer the *maximum compression* experienced during the impact interaction, which occurs when the compression velocity equals zero. For $v = 0$, Equation (2.3) becomes

$$x_{\max} = x(0) = \left[\frac{m(\alpha+1)}{k\mu^2} \cdot \left(\mu v_{\text{in}} + \log \left| \frac{1}{1 + \mu v_{\text{in}}} \right| \right) \right]^{\frac{1}{\alpha+1}} . \tag{2.4}$$

As remarked by Marhefka and Orin [28], Equation (2.1) together with figure 2.1 show that the force f becomes sticky (inward) when $v < v_{\text{lim}} \stackrel{\triangle}{=} -1/\mu$. However there is no physical inconsistency in this "stickiness" property, and indeed this never occurs for a trajectory with initial conditions $x(0) = 0$ and $\dot{x}(0) = v_{\text{in}}$.

Finally, by substituting Equation (2.3) in (2.2) it is possible to plot the compression-force characteristics during collision, which is shown in figure 2.2. It can be noted that the dissipative term $\lambda x^\alpha v$ introduces hysteresis around the curve kx^α.

Output velocity The *restitution coefficient* E is defined as

$$E \stackrel{\triangle}{=} \left| \frac{v_{\text{out}}}{v_{\text{in}}} \right| . \tag{2.5}$$

In this connection a remark can be made: v_{in} and v_{out} correspond to the roots of the right-hand side of Equation (2.3), that is the points where $x = 0$. As a result, v_{out} can be defined implicitly from Equation (2.3) as

$$\mu v_{\text{out}} - \log |1 + \mu v_{\text{out}}| = \mu v_{\text{in}} - \log |1 + \mu v_{\text{in}}| \tag{2.6}$$

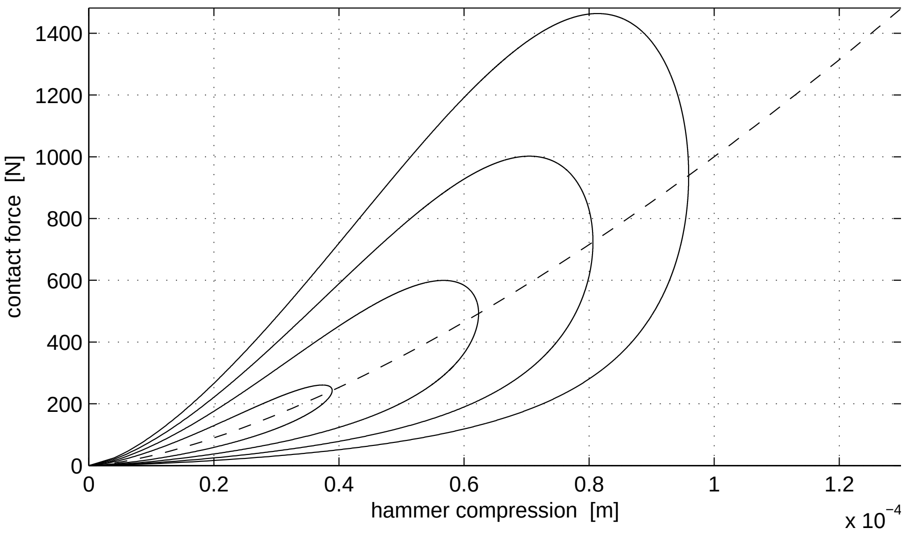


Figure 2.2: Compression-force characteristics. Solid lines represent the case when dissipation is taken into account (the values of parameters are the same as in figure 2.1). The dashed line represents the case when $\lambda = 0$, that is no dissipation is considered.

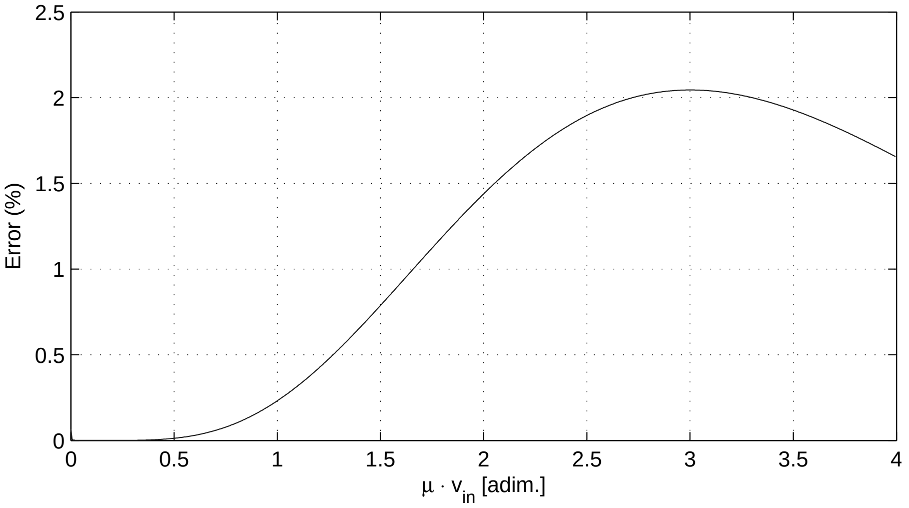


Figure 2.3: Percentage error introduced by Equation (2.7) and (2.8).

where v_{out} is defined as a function of (μ, v_{in}). This implies that μv_{out} is a function of μv_{in} only, and therefore E is also a function of μv_{in} only. However, analytical derivation of the dependence $E(\mu v_{\text{in}})$ have been classically performed in the limit of small initial velocities and/or small dissipation [24].

A non-local approximation for v_{out} can be empirically determined by fitting the curve $E(\mu v_{\text{in}})$ in the two limit regions $\mu v_{\text{in}} \to 0$ and $\mu v_{\text{in}} \to \infty$, giving

$$v_{\text{out}}(\mu, v_{\text{in}}) = v_{\text{lim}} \left[1 - \left(\sum_{j=0}^{n} b_j \cdot v_{\text{in}}^j \right) e^{-2\mu v_{\text{in}}} \right], \tag{2.7}$$

where, in the case $n = 4$, the coefficients b_j are

$$b_0 = 1, \quad b_1 = \mu, \quad b_2 = \frac{2}{3}\mu^2, \quad b_3 = \frac{2}{9}\mu^3, \quad b_4 = \frac{14}{135}\mu^4. \tag{2.8}$$

Figure 2.3 shows the error introduced by Equation (2.7) for $n = 4$ and the coefficients (2.8), when compared to the corresponding value computed numerically as a zero of Equation (2.6).

Contact time It is shown in [4] that the contact time can be expressed as

$$\begin{aligned}\tau = &\left(\frac{m}{k}\right)^{\frac{1}{\alpha+1}} \cdot \left(\frac{\mu^2}{\alpha+1}\right)^{\frac{\alpha}{\alpha+1}} \cdot \\ &\cdot \int_{v_{\text{out}}}^{v_{\text{in}}} \frac{1}{(1+\mu v)\left[-\mu(v-v_{\text{in}}) + \log\left|\frac{1+\mu v}{1+\mu v_{\text{in}}}\right|\right]^{\frac{\alpha}{\alpha+1}}}.\end{aligned} \tag{2.9}$$

Equation (2.9) states that the contact time τ depends only on μ, the exponent α, and the ratio m/k, plus obviously the impact velocity v_{in}. Since neither m nor k affect the value of the integral (recall that v_{out} depends only on μ and v_{in}), it follows that, given a fixed v_{in}, the power-law dependence $\tau \sim (m/k)^{1/(\alpha+1)}$ holds.

From an auditory point of view, the value of the contact time is strongly correlated to the perceived "hardness" of the impact [4]. Namely, as the contact time decreases, the perceived hardness increases. Recalling the power-law dependence above and (2.1) it follows that, for a fixed mass m, "hard" and "soft" impacts correspond respectively to high and low force values.

Energy properties and behavior The energy variation in a mechanical system can be calculated as the work made by the overall force f acting on the system along a certain path $x_1 \rightarrow x_2$:

$$\Delta H = \int_{x_1}^{x_2} f(x)\ dx = \int_{t_1}^{t_2} f(t)v(t)\ dt \tag{2.10}$$

where H is the total energy content, known as the *Hamiltonian*, and the second integral is obtained by considering that t_1 and t_2 correspond respectively to the instants when the displacements x_1 and x_2 are reached. The Hamiltonian H is the sum of potential and kinetic energies (named V and T, respectively):

$$H(t) = V(t) + T(t). \tag{2.11}$$

With regard to the system represented by Equation (2.2), T is related to the dynamics of the point-mass, which is described by the left-hand side of Equation (2.2), while V is related to the elastic component of the impact force given in Equation (2.1).

More specifically, and in agreement with the last integral in Equation (2.10), multiplying both sides of Equation (2.2) by $v(t) = dx/dt$ and time-integrating them, gives

$$\underbrace{\int_0^t ma(s)v(s)\ ds}_{T(t)-T_0} = -\underbrace{\int_0^t kx(s)^\alpha v(s)\ ds}_{V(t)} - \underbrace{\int_0^t \lambda x(s)^\alpha v(s)^2\ ds}_{\Lambda(t)>0} \tag{2.12}$$

where the force expression given in Equation (2.1) has been considered in the case $x > 0$ only, and $\Lambda(t)$ is the work made by the dissipative component of the impact force. The integrals $V(t)$ and $T(t)$ in Equation (2.12) can be solved explicitly, giving

$$V(t) = \frac{kx(t)^{\alpha+1}}{\alpha+1}, \quad T(t) = \frac{mv(t)^2}{2}. \tag{2.13}$$

Now consider a system where the point-mass travels with velocity v_{in} before the impact occurs, then the initial Hamiltonian corresponds to the initial kinetic energy and is

$$H_0 = T_0 = \frac{mv_{\text{in}}^2}{2}. \tag{2.14}$$

At every time instant t, the overall variation of energy is:

$$\Delta H(t) = H(t) - H_0 = -\Lambda(t), \tag{2.15}$$

and therefore the total amount of energy dissipation occurred during the impact interaction is

$$\Delta H_\tau = H_\tau - H_0 = -\Lambda_\tau, \tag{2.16}$$

where τ indicates the instant when the impact ends, and H_τ is the final Hamiltonian of the system, that is the energy content after the impact interaction has ended, which equals:

$$H_\tau = T_\tau = \frac{mv_{\text{out}}^2}{2}. \tag{2.17}$$

It can be noted that ΔH_τ corresponds to the area enclosed by the hysteresis loops represented in Figure 2.2.

As a last remark, from the results above it clearly follows that

$$0 \leq H(t + dt) \leq H(t). \tag{2.18}$$

2.1.2 Numerical simulations

In this section, the continuous-time system described by Equation (2.2) is discretized by means of several numerical methods, and the resulting numerical systems are studied.

Remarks on accuracy and stability

All numerical systems can simulate their continuous-time counterparts only to some extent. Generally speaking, one of the basic reasons for this inherent limit is that the discretized variables depend on the chosen *sample rate* F_s, and therefore the behavior of any numerical method is bounded by it.

For instance, since our numerical system takes as input the initial velocity v_{in} of the point-mass, the compression x can only be computed as a function (integral) of ${v_{\text{in}}}^2$. It is clear that, much as good a numerical method can be, the higher v_{in} and/or the lower F_s (i.e., less samples are available for computing the numerical integral) are, the less precise x is. In any case, the computed compression is *always* an approximate value.

Besides, the resulting impact force f is obviously affected too, and as the impact "hardness" increases, the computation error increases as well. From a more general perspective, those inconsistencies are reflected in the energy behavior of the numerical system: for example, it is evident that inconsistently large compressions cause the force, the output velocity, and therefore the energy, to increase inconsistently with the continuous-time system.

Apart from the intrinsic numerical bounds pointed above, it must be noted that different numerical methods behave differently, and that *all* of them are susceptible to problems related to stability. For instance, for some values of parameters, one method could dissipate too much energy, while another one could violate the principle of passivity, thus generating spurious energy. In both cases those inconsistencies - besides the fact of resulting in an erroneous state of the system - could give birth to instabilities.

Even though some stability conditions can be derived for LTI systems[3], those do not extend to the case of non-linear systems. In the latter case one possibility is to exploit energy-based methods (see for example [5] in the context of physical modeling sound synthesis).

[2] It is clear that, while the dependence on the impact velocity has a physical meaning, the dependence on the sample rate has not.

[3] Consider, for example, the von Neumann analysis [41].

Numerical methods

Hereafter, the continuous-time system of Equation (2.2) is discretized using different numerical methods. Following the standard notation in numerical mathematics, the integration step is a constant named h.

1-step Adams-Moulton The *1-step Adams-Moulton* (AM1) [41] is a A-stable 2nd-order implicit method, also known as *bilinear transformation*, or *trapezoidal rule.*

Discretizing the equation of motion (left-hand side of Equation (2.2)) results in the following state-space form equation:

$$\begin{bmatrix} x_{n+1} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ v_n \end{bmatrix} + \begin{bmatrix} \frac{h^2}{4m} \\ \frac{h}{2m} \end{bmatrix} [f_{n+1} + f_n]. \tag{2.19}$$

where the expression for the discrete-time force can be obtained by replacing the continuous-time variables $x(t)$ and $v(t)$ in Equation (2.1) with their discrete-time counterparts.

Since the AM1 method is implicit, Equation (2.19) is also in implicit form and this is reflected in the instantaneous relationship between $[x_{n+1}\ v_{n+1}]^T$ and f_{n+1}. Unfortunately, since f_{n+1} also has an instantaneous dependence on x_{n+1} and v_{n+1}, the discrete-time counterpart of the system described by Equation (2.2) contains a delay-free loop, which is not directly computable and – because of the included non-linearities – needs some special handling in order to be solved. In particular, the *K-method* [7] together with *Newton's method* [41] are used, weighing on the efficiency of the simulation.

Velocity Verlet The *Verlet method* [49] is a 2nd-order explicit method which is commonly used in computer graphics [33], video games, and molecular dynamics simulation. Its main application is that of integrating Newton's equation of motion in order to describe the trajectory of moving particles. The one used here is a variant called *velocity Verlet*, which provides better handling of the velocity variable and can be seen as a predictor-corrector method.

Discretizing the system represented in Equation (2.2), results in the following implementation scheme:

$$\begin{aligned} x_{n+1} &= x_n + hv_n + \frac{h^2}{2}\frac{f_n}{m} \\ v_{n+\frac{1}{2}} &= v_n + \frac{h}{2}\frac{f_n}{m}, \quad \text{predictor} \\ f_{n+1} &= f(x_{n+1}, v_{n+\frac{1}{2}}), \\ v_{n+1} &= v_{n+\frac{1}{2}} + \frac{h}{2}\frac{f_{n+1}}{m}, \quad \text{corrector.} \end{aligned} \tag{2.20}$$

It should be noted that this algorithm assumes that f_{n+1} only depends on the predicted velocity $v_{n+\frac{1}{2}}$, which clearly gives rise to inaccuracies.

Heun The *Heun method* [41] is a predictor-corrector explicit method, with the forward Euler method as predictor and the trapezoidal rule as corrector. It can also be seen as a *2nd-order Runge-Kutta method* (RK2).

Discretizing Equation(2.2) results in the following implementation scheme:

$$\begin{aligned}
\tilde{v}_{n+1} &= v_n + h\frac{f_n}{m}, \quad \text{predictor} \\
x_{n+1} &= x_n + \frac{h}{2}(v_n + \tilde{v}_{n+1}) \\
f_{n+1} &= f\left(x_{n+1}, \tilde{v}_{n+1}\right), \\
v_{n+1} &= v_n + \frac{h}{2}\frac{f_n + f_{n+1}}{m}, \quad \text{corrector.}
\end{aligned} \tag{2.21}$$

Again, it should be noted that both x_{n+1} and f_{n+1} only depend on the predicted velocity $\tilde{v}_{n+1}$, and this gives rise to inaccuracies.

4th-order Runge-Kutta The *4th-order Runge-Kutta* [41] is an explicit iterative method which is widely used to solve ODEs with improved accuracy.

Discretizing Equation (2.2) results in the following implementation scheme:

$$\begin{aligned}
x_{n+1} &= x_n + \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4) \\
v_{n+1} &= v_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned} \tag{2.22a}$$

where

$$\begin{aligned}
l_1 = hv_n, &\quad l_2 = h(v_n + \frac{k_1}{2}), \\
l_3 = h(v_n + \frac{k_2}{2}), &\quad l_4 = h(v_n + k_3) \\
k_1 = h\frac{f_n}{m}, &\quad k_2 = h\frac{f(x_n + \frac{l_1}{2}, v_n + \frac{k_1}{2})}{m}, \\
k_3 = h\frac{f(x_n + \frac{l_2}{2}, v_n + \frac{k_2}{2})}{m}, &\quad k_4 = h\frac{f(x_n + l_3, v_n + k_3)}{m}.
\end{aligned} \tag{2.22b}$$

It should be noted that, for each sample, both the velocity and the non-linear force given in Equation (2.1) need to be evaluated four times, therefore affecting the efficiency of the simulation.

Experimental results

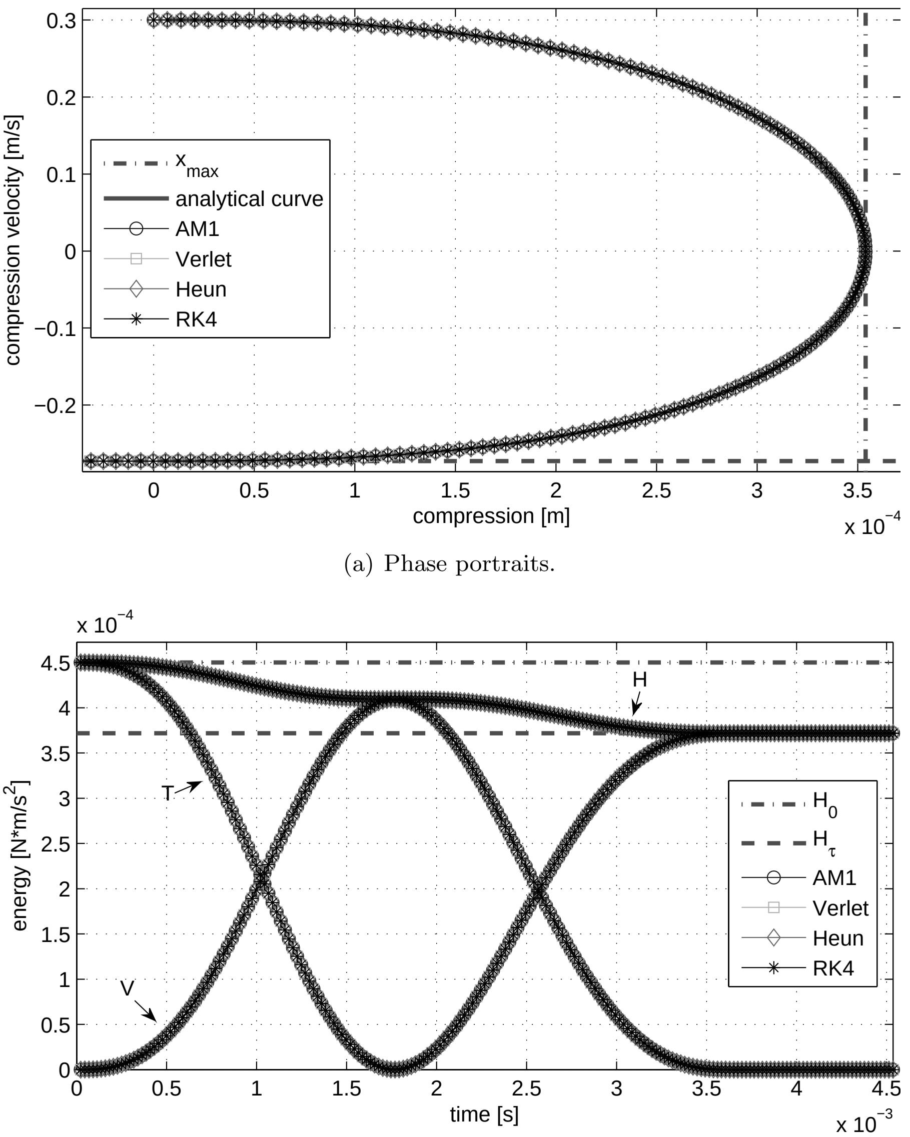
In order to evaluate the chosen numerical methods, it is useful to compare the behavior of the corresponding simulations against the known analytical results (cfr. section 2.1.1).

In particular, several plots are provided which show: the phase portrait on the (x, v) plane (cfr. figure 2.1), the compression-force characteristic (cfr. figure 2.2) and, in the discrete-time domain, the impact force f and the Hamiltonian H.

The following values of parameters are kept constant throughout the simulations: $m = 10^{-2}$ kg, and $F_s = 44.1$ kHz, that is a standard audio sample rate. The considered integration step h is therefore equal to $1/F_s$.

Reference simulations: Soft impact In order to verify the simulations and provide a reference, the model's parameters are set to a "soft" configuration where the impact interaction extends over many samples, so that the numerical systems operate in a safe space and should therefore behave very similarly to the original continuous-time system.

This is confirmed by figure 2.4, where the plots of all the simulations substantially overlap. In particular, it is possible to check the accuracy of the simulations against the analytical results provided in section 2.1.1: figure 2.4(a) displays the phase portrait on the (x, v) plane and, as tangent lines, the



(a) Phase portraits.

(b) Energy behaviors.

Figure 2.4: Comparison of different methods in a soft impact case. The values of parameters are $k = 10^6$ N/m$^\alpha$, $\mu = 0.5$ s/m, $\alpha = 1.6$, $v_{\rm in} = 0.3$ m/s.

maximum compression $x_{\max}$ and the output velocity $v_{\rm out}$, while in figure 2.4(b) two horizontal lines display $H_0 = T_0$ and $H_\tau = T_\tau$, that is respectively the initial and the final Hamiltonian.

Figure 2.4(b) also allows to understand the energy behavior of the system: the sinusoid-like curves represent the kinetic energy T (curve starting at $H_0 = T_0$) and the elastic potential energy V (curve starting at 0). The sum of the potential and kinetic energies provides the upper staircase-like curve representing the Hamiltonian H, which indeed decreases[4] until $H_\tau = T_\tau$ in correspondence to the end of the contact interaction.

General study It can be observed that for hard impacts (that is when the contact time lasts only a few samples) and/or when $\mu v_{\rm in} \to 0_+$, the energy behavior of some numerical implementations becomes inconsistent with the continuous-time system.

The main reference used to qualitatively and/or quantitatively assess the reliability of a particular numerical method is the energy behavior of the corresponding simulation. In particular, the following

[4] When $\mu = 0$ the overall energy remains constant, that is $H = H_0 = H_\tau$.

condition

$$H_{n+1} \leq H_n, \quad \text{for all } n \tag{2.23}$$

should be satisfied (see Equation (2.18)). Other indicators are provided by the analytical values of maximum compression x_{max} and output velocity v_{out}, which should never be exceeded.

It is found empirically that when $\tau \leq 4$ samples, the percentage errors quickly increase, and the reliability of all the simulations is poor. Hence, for the sake of clarity, in the study hereafter only values of parameters resulting in $\tau > 4$ samples are considered.

Case 1: $\mu v_{\text{in}} \to 0_+$

Even in case of "not-too-hard" impacts[5], the Hamiltonian of both Verlet- and Heun-discretized systems is prone to oscillations, and the system generally results in an inconsistent final energy state (typically, $H_\tau^{\text{Verlet, Heun}} > H_\tau$). Moreover, as put forward in section 2.1.2, for hard impacts (high k and/or low α) the maximum compression is often higher than the analytical value x_{max}. As for AM1-discretized systems, these generally tend to dissipate too much energy during the compression phase, while gaining spurious energy during the decompression phase (i.e. $H_\tau^{\text{AM1}} > H_\tau$). On the other hand, RK4-discretized systems generally behave quite consistently (i.e. $H_\tau^{\text{RK4}} \approx H_\tau$).

Figure 2.5 shows an example simulation of hard impact (contact time = 8 samples) with $\mu v_{\text{in}} = 0.03$, while table 2.1(a) shows the corresponding percentage errors.

Case 2: hard impacts with average or high values of μv_{in}

During the impact interaction, all the simulations usually result in a sufficiently consistent time evolution of the Hamiltonian: the condition (2.23) is generally satisfied, except for the end of the interaction.

At this stage, the two 2nd-order explicit methods typically tend to introduce spurious energy (i.e. $H_\tau^{\text{Verlet, Heun}} > H_\tau$). Concerning AM1-discretized systems, these tend to behave better than the two methods mentioned above, even if sometimes $H_\tau^{\text{AM1}} > H_\tau$. Again, RK4-discretized systems generally behave rather consistently, and only experience small discrepancies with respect to the analytical results (i.e. $x_{\text{max}}^{\text{RK4}} \approx x_{\text{max}}$ and $H_\tau^{\text{RK4}} \approx H_\tau$).

A second hard impact example simulation is provided, following the values of parameters adopted in figure 2.1 for $v_{in} = 1$ m/s ($\mu v_{\text{in}} = 0.5$), while table 2.1(b) shows the corresponding percentage errors. The resulting contact time τ equals 6 samples.

(a) Simulation example following *case 1*. The values of parameters are the same as in figure 2.5.

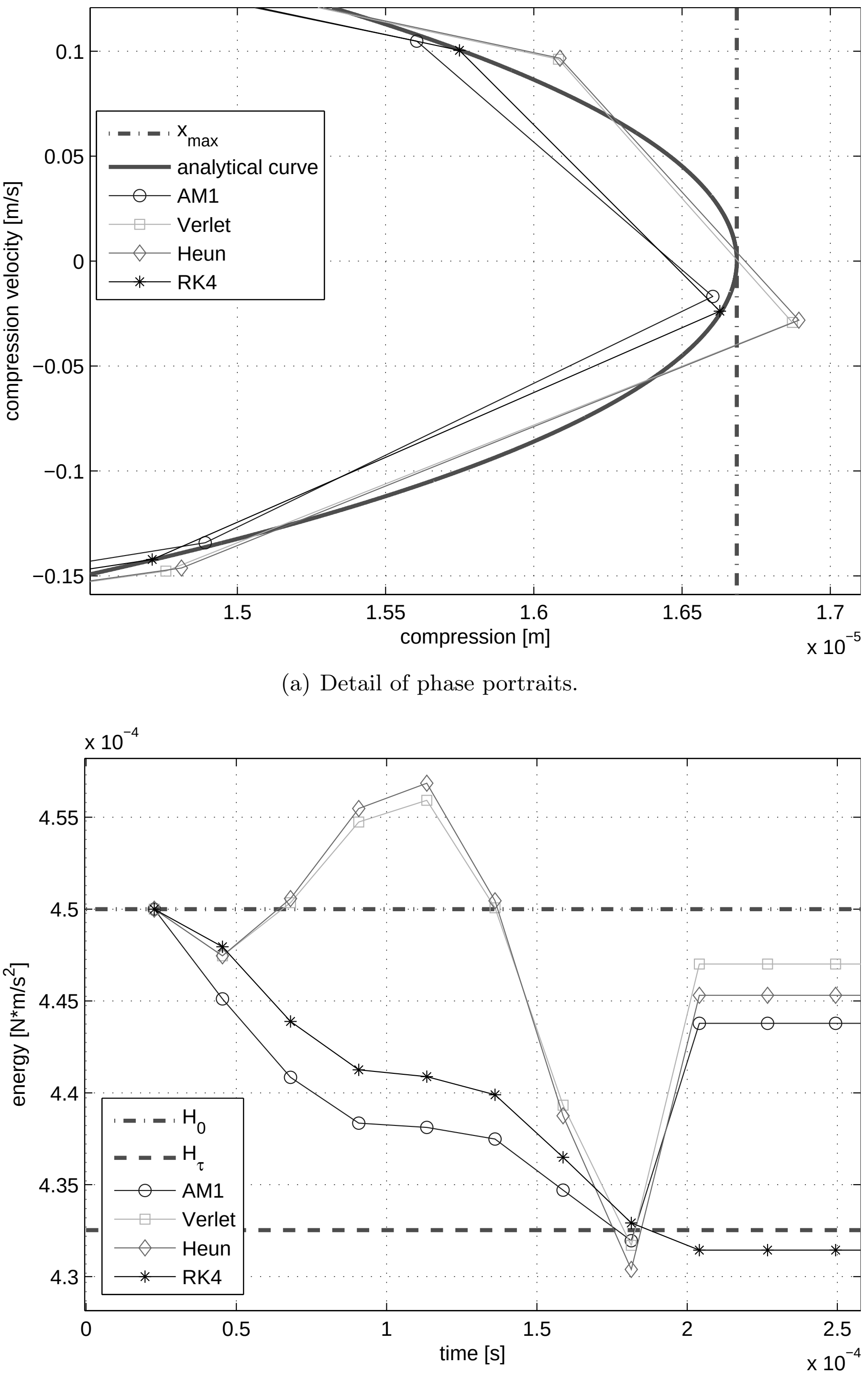
%err	AM1	Verlet	Heun	RK4	num. res.
x_{max}	n.e.	+1.122	+1.254	n.e.	-
v_{out}	+1.293	+1.660	+1.467	−0.125	$\sim -10^{-2}$
H_τ	+2.603	+3.348	+2.955	−0.250	-

(b) Simulation example following *case 2*. The values of parameters are the same as in figure 2.1 for $v_{\text{in}} = 1$ m/s.

%err	AM1	Verlet	Heun	RK4	num. res.
x_{max}	n.e.	n.e.	n.e.	n.e.	-
v_{out}	+2.551	+0.839	−4.692	−0.105	$\sim -10^{-4}$
H_τ	+5.166	+1.685	−9.164	−0.211	-

Table 2.1: Summary of percentage errors for two hard-impact example simulations. The caption "n.e." stands for "not exceeded". The last column shows the error resulting from comparing the approximate value v_{out}, which is obtained from Equations (2.7) and (2.8), against a value computed numerically as a zero of Equation (2.6) (cfr. figure 2.3).

[5] e.g., when the contact time lasts tens of samples

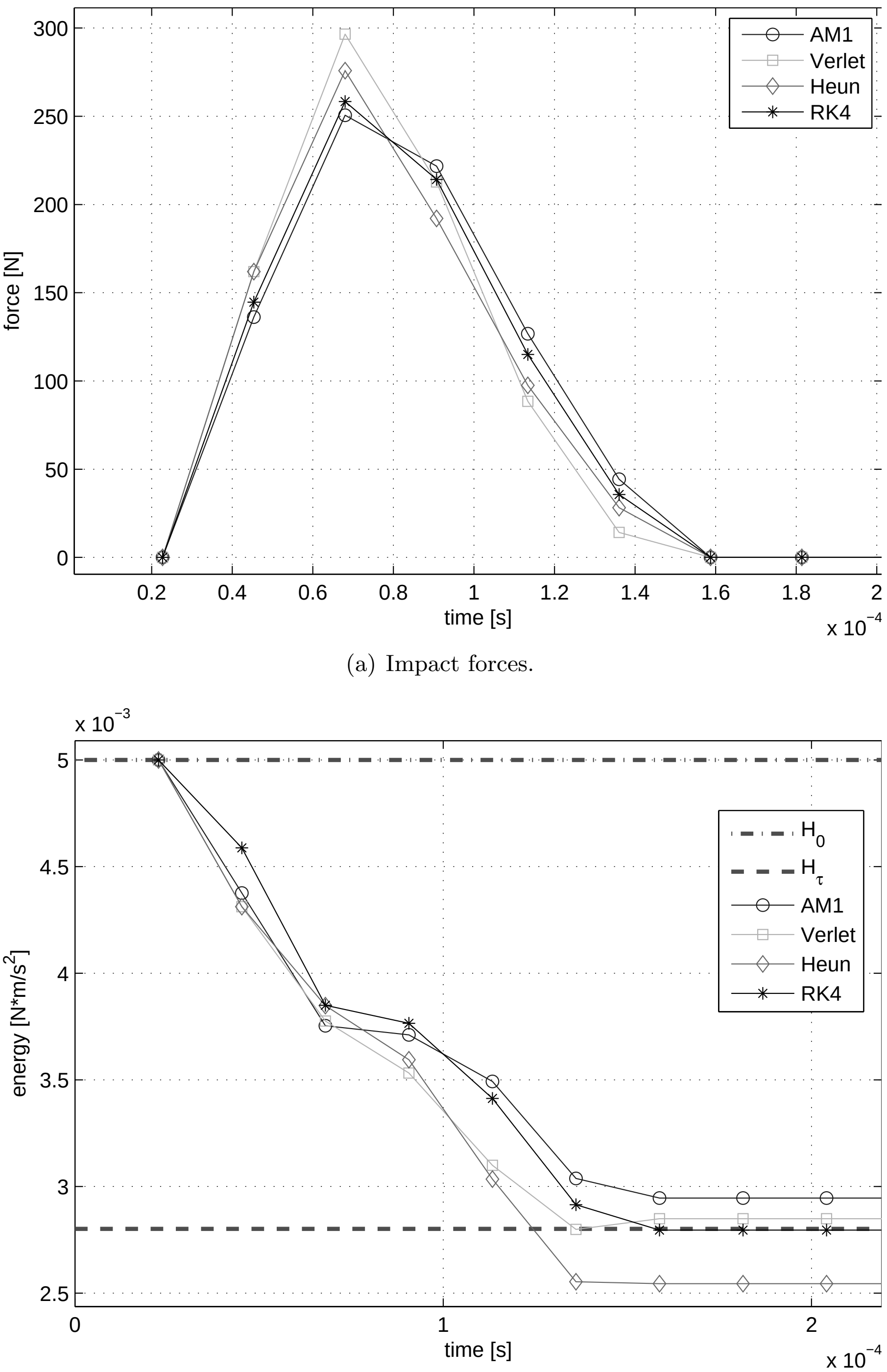


(a) Detail of phase portraits.

(b) Detail of energy behaviors.

Figure 2.5: Hard impact example following *case 1*: Comparison of different numerical methods. The values of parameters are $k = 10^7$ N/m$^\alpha$, $\mu = 0.1$ s/m, $\alpha = 1.1$, $v_{\text{in}} = 0.3$ m/s. The contact time equals 8 samples.

From the simulations above it is clear that the percentage error is not strictly correlated to the impact hardness (that is, the duration in samples of the contact interaction). Moreover, the nonlinearities of the impact model described by Equation (2.1) make it difficult to predict the exact behavior of the corresponding numerical system, especially with lower-order explicit methods.



(a) Impact forces.

(b) Detail of energy behaviors.

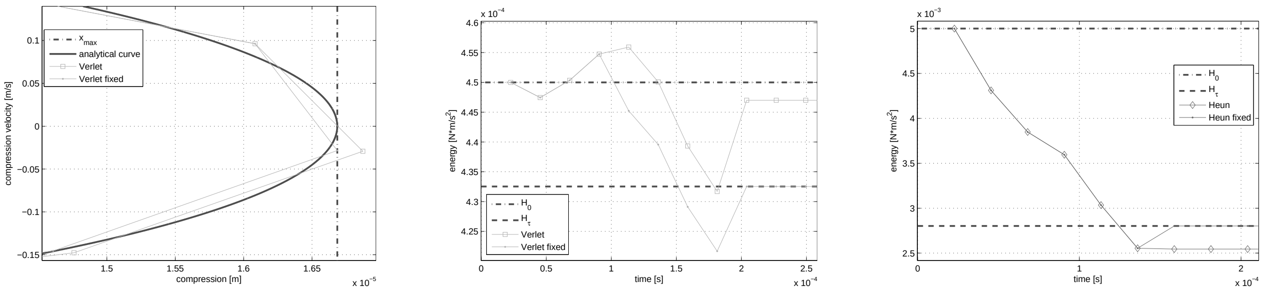
Figure 2.6: Hard impact example following *case 2*: Comparison of different numerical methods. The values of parameters are the same as in figure 2.1 with $v_{\mathrm{in}} = 1$ m/s. The contact time equals 6 samples.

2.1.3 Improved numerical simulations

Exploitation of analytical results

In this section, some solutions are proposed to partially fix the inconsistencies pointed out in section 2.1.2. The aim is to improve the reliability of simulations which use the impact model under study, in view of their implementation as real-time applications.

As written above, in order for the numerical systems to behave consistently, the energy condition



(a) *Case 1*: phase portrait of the Verlet-discretized system. The compression x is forced to $x_{\max}$.

(b) *Case 1*: energy behavior of the Verlet-discretized system. During the interaction, the Hamiltonian is influenced by the correction of x, and as the interaction ends, the velocity is forced to v_{out}, resulting in $H = H_\tau$.

(c) *Case 2*: energy behavior of the Heun-discretized system. As the interaction ends, the velocity is forced to v_{out}, resulting in $H = H_\tau$.

Figure 2.7: Comparison of simulations with and without corrections.

given in Equation (2.23) must hold. Moreover, from figure 2.1 it can be inferred that the relation $v(t + dt) \leq v(t)$ holds. This brings us to the following condition:

$$v_{n+1} \leq v_n, \quad \text{for all } n. \tag{2.24}$$

However, at the time of this writing no analytical result describing the behavior of the state variables $x(t)$ and $v(t)$ over time is available, and therefore there is no easy option that allows to fix wrong numerical values of H or v *during* the impact interaction.

One indirect approach is to intervene on the compression, for which the following condition must hold:

$$x_n \leq x_{\max}, \quad \text{for all } n, \tag{2.25}$$

where $x_{\max}$ is a constant calculated by Equation (2.4). Hence the suggested solution is that of clipping x_n to the maximum allowable value $x_{\max}$ whenever this is exceeded. This also affects the Hamiltonian of the system by decreasing the elastic potential energy V_n.

An additional solution is to force the numerical output velocity to the approximate value v_{out} calculated by Equations (2.7) and (2.8). As a result, when the interaction ends, the energy content of the numerical system is restored. That is, the Hamiltonian is forced to H_τ. Since the error introduced by the approximate value of v_{out} – even in case of hard impacts – is generally of order lower than that of the error introduced by the numerical simulations (cfr. table 2.1), the correction of v_{out} ensures that the physical consistency of the numerical system is improved.

In order to implement the suggested corrections, it is necessary to compute the values of $x_{\max}$ and v_{out}, which are given respectively by Equations (2.4), and (2.7) plus a nth-order expansion for the coefficients b_j (e.g., the 4th-order expansion (2.8)). These computations only need to take place in correspondence of an impact event, and more precisely as soon as the impact velocity v_{in} is known.

Numerical simulations with corrections

As a basic application example, the worst behaving impact simulations found in *case 1* and in *case 2* of section 2.1.2 were corrected using the solutions explained above. Figure 2.7 shows a comparison of the simulations with and without corrections. In the corrected simulations, the compression is limited to $x_{\max}$ and the energy content is forced to H_τ upon the end of the interaction.

Furthermore, in order to verify the necessity of corrections even in case of small errors, an external force was applied to the point-mass, this way simulating a bouncing object. It is found that when the point-mass travels with even slightly wrong velocities, the trajectories are inaccurate, the rebounds

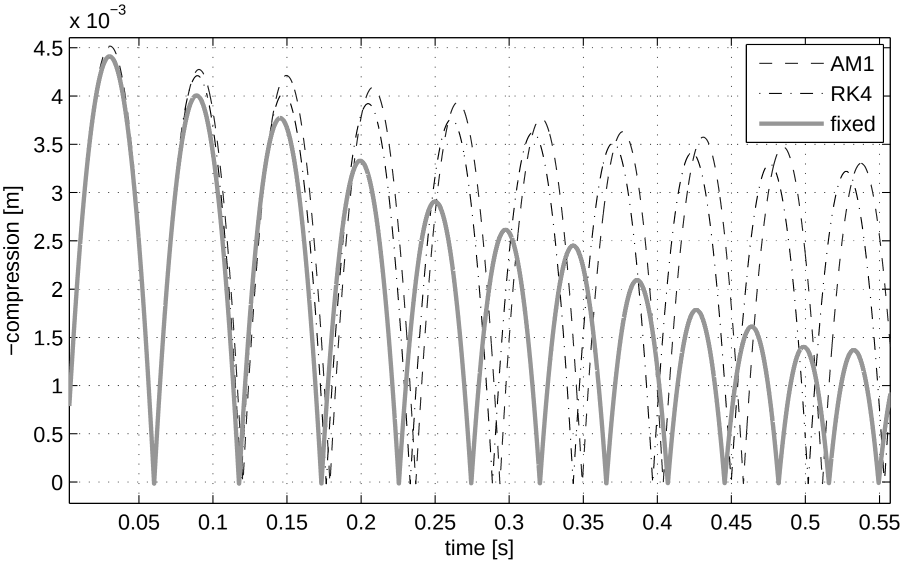


Figure 2.8: Effect of errors when the force of gravity is applied to the simulation example of *case 1*. For the sake of clarity, the sign of the compression has been inverted, this way visually representing the rebounds as they would occur on a horizontal floor.

happen at the wrong time, energy is wrongly dissipated, and above all those errors accumulate at each impact event. Figure 2.8 shows the effect of small errors in the situation pointed above, where the force of gravity $f_g = mg$ is superimposed on the impact simulation example provided in section 2.1.2 and corresponding to *case 1*. Note that, in order to stress that even small errors can strongly affect the physical consistency, only the two *best* performing simulations (namely, AM1 and RK4) are portrayed.

2.1.4 Ongoing work

A new correction method is being developed which allows to accurately control the dynamical system during the impact interaction and upon the detachment. This a major improvement over the current method, which only partially address the control of the system.

Moreover, the research is currently being extended to the cases where vibrational losses are present (that is resonating objects) and when external forces are applied, this way improving both the stability and the versatility of the algorithm for sound synthesis and haptic rendering.

Finally, a modified impact model is under study whose purpose is to simulate impact distributed in both space and time, this way allowing to better simulate the impact between surfaces.

2.2 Friction modeling

Friction sounds are another important category of sounds and a basic building block which can be used in the simulation of several walking events. As an example, friction is present when sliding a foot on a floor or when walking on a creaking surface. In order to adequately describe friction phenomena at low velocities (i.e., stick-slip motion, presliding behavior, frictional memory, etc.) researchers have recently developed dynamic models that describe the dependence of friction f on the relative velocity v between two contacting bodies through a differential equation [16].

Dynamic models are able to take into account presliding behavior for very small displacements, where the friction force increases gradually with the displacement.

2.2.1 The friction model

Assuming that friction results from a large number of microscopic elastic bonds, called bristles, the v-to-f relationship is expressed as:

$$f(z, \dot{z}, v, w) = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 v + \sigma_3 w \tag{2.26}$$

where z is the average bristle deflection, the coefficient σ_0 is the bristle stiffness, σ_1 the bristle damping, and the term $\sigma_2 v$ accounts for linear viscous friction. The fourth component $\sigma_3 w$ relates to surface roughness, and is simulated as fractal noise.

The term $w(t)$ in the fourth component of equation (2.26) is a pseudo-random function of time which introduces noise in the force signal, and is therefore related to surface roughness. Such a term can be rendered as a fractal process, as it will be illustrated in section 2.2.2. Indeed fractal processes are widely used in computer graphics, since they provide surfaces and textures that look natural to a human eye. Since in physics-based modeling there is direct translation of geometric surface properties into force signals and, consequently, into sound, it seems natural to follow the same fractal approach to surface modeling.

2.2.2 Adding fractal noise

In the remainder of this section we address the problem of surface texture rendering by means of fractal processes.

Fractals are generally defined as scale-invariant geometric. They are self-similar if the rescaling is isotropic or uniform in all directions, self-affine if the rescaling is anisotropic or dependent on the direction, as statistically selfsimilar if they are the union of statistically rescaled copies of themselves.

More formally, a one-dimensional fractal process can be defined as a generalization of the definition of standard Brownian motion. A stochastic process $x = x(t), x \geq 0$ is standard Brownian motion if:

1. the stochastic process x has independent increments;
2. the property $x(t) - x(s) \sim \mathcal{N}(0, kt - s)$ with $0 \leq s \leq t$ holds. That is, the increment $x(t) - x(s)$ is normally distributed with mean 0 and variance equal to $(t - s)$;
3. $x(0) = 0$.

The definition of standard Brownian motion can be generalized to the definition of fractal process, if the increment $x(t) - x(s)$ is normally distributed with mean 0 and variance proportional to $(t - s)^{2H}$. The parameter H is called Hurst exponent, and characterizes the scaling behaviour of fractal processes: if $x = x(t), x \geq 0$ is a fractal process with Hurst exponent H, then, for any real $a > 0$, it obeys the scaling relation

$$x(t) \overset{P}{=} a^{-H} x(at)$$

where $\overset{P}{=}$ denotes equality in a statistical sense. This is the formal definition for statistical self-similarity. The $1/f$ family of statistically self-similar processes, also known as $1/f$ noise, is defined as having power spectral density $S_x(\omega)$ proportional to $1/\omega^\beta$ for some spectral parameter related to the Hurst exponent H by $\beta = 2H + 1$. For $\beta = 0$ the definition corresponds to white noise, for $\beta = 2$ Brownian noise is obtained, and for $\beta = 1$ the resulting noise is referred to as pink noise.

The parameter β also relates to the fractal dimension. The fractal dimension of a function is related to the roughness of its plot and is exploited in computer graphics to control the perceived roughness. For $1/f$ processes, it is inversely proportional to the Hurst exponent H. Larger values of H correspond to lower values of the fractal dimension and H is proportional to β. Therefore, by increasing β, we will achieve a redistribution of power from high to low frequencies, with an overall smoothing of the waveform.

The problem of generating $1/f$ noise has been treated extensively. One of the most common approaches amounts to properly filtering a white noise source in order to obtain a $1/f$ spectrum. We follow here this approach. The shaping filter is a cascade of N first-order filters, each with a real zero-pole pair. The overall transfer function $H(s)$ in the Laplace domain is the following:

$$H(s) = A\frac{\prod_{i=1}^{N}(s - s_{zi})}{\prod_{i=1}^{N}(s - s_{pi})} \tag{2.27}$$

where A is a suitable constant.

The fractal noise generator is obtained by properly setting the poles and the zeros of the filters in the cascade. Specifically, the pole and zero frequencies, f_{pi} and f_{zi}, can be computed as functions of the spectral slope β with the following formulas:

$$f_{pi} = -\tfrac{s_{pi}}{2\pi} = f_{p(i-1)}10^{\frac{1}{h}}$$
$$f_{zi} = -\tfrac{s_{zi}}{2\pi} = f_{pi}10^{\frac{\beta}{2h}}$$

where f_{p1} is the lowest pole frequency of the filter. Therefore, the lowest limit of the frequency band for the approximation is f_{p1} and the range width is expressed in decades. The density h (density of the poles per frequency decade) can be used to control the error between the target spectrum and the approximated spectrum obtained by white noise filtering.

The transfer function in the discrete-time domain can be computed with the Impulse Invariant method. It is known that this corresponds to mapping poles and zeros of the transfer function $H(s)$ to poles and zeros of the transfer function $H(z)$ in the discrete-time domain by making the following substitution: $s - s_x \to 1 - e^{s_x T_s} z^{-1}$ where T_x is the sampling period and s_x stands for a pole s_{pi} or a zero s_{zi}. The following discrete transfer function is then obtained:

$$H(z) = A'\frac{\prod_{i=1}^{N} 1 - e^{s_{zi}T} z^{-1}}{\prod_{i=1}^{N} 1 - e^{s_{pi}T} z^{-1}} \tag{2.28}$$

where A' is a normalizing constant. In conclusion, the $1/f$ spectrum is approximated by a cascade of first-order filters, each one with the following discrete transfer function:

$$H^{(i)}(z) = \frac{1 + b_i z^{-1}}{1 + a_i z^{-1}}, \tag{2.29}$$

with

$$\begin{cases} a_i = e^{-2\pi f_{pi} T}, & b_i = e^{-2\pi f_{zi} T} \\ f_{pi} = f_{p(i-1)}10^{\frac{1}{h}}, & f_{zi} = f_{pi}10^{\frac{\beta}{2h}} \end{cases}$$

2.2.3 Integration of the fractal noise within the friction model

The fractal noise generator has been integrated within the friction model as a flext external. This allows to control information on surface roughness by controlling the fractal dimension of the noisy component $\sigma_3 w(t)$.

For convenience in implementation, the shaping filters (2.29) are rewritten as a cascade of N/2 second-order filters (biquads).

The most relevant parameter accessible to the user is the roughness β, which defines the target $1/f$ spectrum. The fractal noise gain (σ_3) can be also set, as well as the frequency of the first pole (f_{p1}).

2.2.4 Ongoing work

Figure 2.9 (above) shows the Max/MSP patch enabling the synthesis in real time of sounds obtained by using the integrated friction plus fractal noise model. The algorithm is controlled by the noise gain,

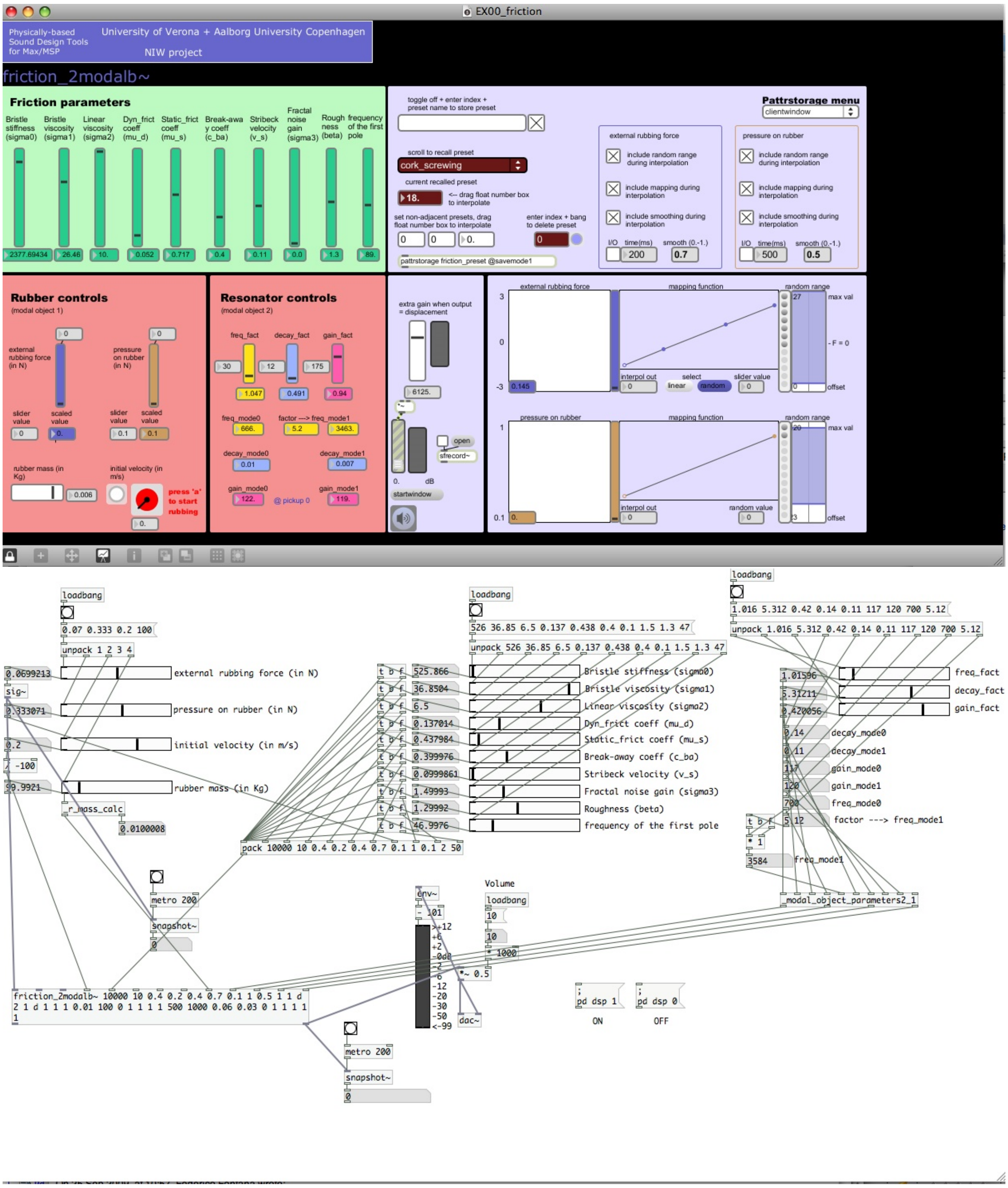

EX00_friction
Physically-based Sound Design Tools for Max/MSP
University of Verona + Aalborg University Copenhagen
NIW project
friction_2modalb~
Friction parameters
Bristle stiffness (sigma0)
Bristle viscosity (sigma1)
Linear viscosity (sigma2)
Dyn_frict coeff (mu_d)
Static_frict coeff (mu_s)
Break-away coeff (c_ba)
Stribeck velocity (v_s)
Fractal noise gain (sigma3)
Roughness (beta)
frequency of the first pole
Pattrstorage menu
cork_screwing
Rubber controls
Resonator controls
external rubbing force
pressure on rubber
mapping function
random range
initial velocity (in m/s)
rubber mass (in Kg)
press 'a' to start rubbing
freq_fact
decay_fact
gain_fact
decay_mode0
decay_mode1
gain_mode0
gain_mode1
freq_mode0
freq_mode1
factor ---> freq_mode1
loadbang
pack 10000 10 0.4 0.2 0.4 0.7 0.1 1 0.1 2 50
friction_2modalb~ 10000 10 0.4 0.2 0.4 0.7 0.1 1 0.5 1 1 d 2 1 d 1 1 1 0.01 100 0 1 1 1 1 500 1000 0.06 0.03 0 1 1 1 1
modal_object_parameters2_1
metro 200
snapshot~
env~
Volume
dac~
pd dsp 1
pd dsp 0
ON
OFF

Figure 2.9: Max/MSP (above) and pd (below) patch of the friction plus fractal noise-based sound engine.

the roughness and the resonant frequency.

This patch is integrated as part of the friction model available on the SDT library. To be compatible both with Max/MSP and puredata, the algorithm has been implemented in flext. The corresponding pd patch can be seen in Figure 2.9 (below).

3 A palette of walking sounds in real-time

This chapter describes the current palette of walking sounds that can be synthesized in real time. Some of these sounds are obtained from physical modeling algorithms implemented in the SDT library, also making use of the friction model enriched by fractal noise—see Chapter 2.

While experimenting with the SDT, the consortium noticed that the strict use of physical models, although promising in the long term, is likely to lead to a reduced sonic palette of material properties in a three-year period. To improve upon this situation, AAU has started to make use of the PhiSM algorithm, that essentially realizes a physically-informed parametric model [12]. The physical and parametric paradigm, once put together, concur to form an interesting palette that will be used in the next project periods.

The same palette has already been partially included in a Max/MSP patch, realizing the first version of a walking sound synthesizer. Once this synthesizer will be coherently integrated as part of the SDT library, a walking sound design tool will become available whose power ultimately depending on the accuracy of sound and its flexibility in becoming part of a walking interface. This integration task, along with the constant improvement of the sounds especially by means of a design-and-evaluation cycle, is part of the planned activity for the next period.

Audio examples taken from this palette are available in `www.niwproject.eu`.

3.1 SDT

The Sound Design Toolkit (SDT [31]) – a library of ecological sound models maintained by UNIVR – has been extended with new examples for the graphical programming environment puredata [38]. In particular, some existing models have been extended in order to simulate the foot-ground interaction, for instance the displacement over time of a surface (ground) in response to an external excitation (footstep). The resulting signal can the be interpreted as

- a source of air pressure perturbation, resulting in **sound synthesis**;
- vibrations, resulting in **haptic rendering**.

Installing the SDT The SDT library is being maintained making use of the version control system *Subversion*[1] (svn).

In order to first download (or check-out, in svn language) a copy of the library it is necessary to have a local installation of a svn client (e.g., it is already included in Mac OSX systems). Go to the command shell and write[2]:

```
svn co https://svn.sme-ccppd.org/svn/sobs/SoundDesignTools/branches/SObWay
```

Once the check-out is completed, you will find a fresh SDT distribution inside your local directory `SObWay`.

In order to update your local copy of the SDT it is sufficient to open a command shell, go inside the SDT directory and write:

```
svn up
```

[1] `http://svnbook.red-bean.com/`

[2] There are also very convenient GUI-based svn clients like `http://tortoisesvn.net/` for Windows.

This way any changes submitted by the developers will be reflected in your local copy.

For the less experienced users, self-contained and directly downloadable release packages will be made publicly available on the project website from time to time.

For further information on how to use the library please refer to [31].

3.2 SDT-based walking sounds

The new example patches are located in `SDT_patches/for_pd/examples` and make use of some features offered by the extended version of Pure Data[3] which allows e.g. to remotely control the sound models using the OSC protocol[4] and to interface them with an Arduino[5] board via serial (USB) communication.

Two different versions of the patches are provided:

1. a version making use of the OSC protocol, which allows to control the sound synthesis process e.g. from a remote computer or a hardware controller compatible with OSC;

2. a version specifically developed for sonifying the augmented-shoes prototype (NIW shoes, refer to Deliverable 3.1) which has been implemented at UNIVR. This version assumes to being provided with force data coming from the four force sensors embedded in the shoes. Also, the audio output is provided as a "2.2" system (two tweeters and two woofers) with adjustable crossover frequency: the first two channels correspond to the tweeters, while the third and fourth channels provide the low-frequency signal for the woofers.

3.2.1 Soft impact

The *soft impact* example patches make use of an existing model simulating the impact between two resonating objects [3, 46]: in detail, in this version one of the resonators is excited by a *noise burst*, which is interpreted as a force signal and simulates the sequence of microscopic impacts resulting from foot-ground interaction. The other resonator, usually used as an exciter, is not being used.

The resulting signal can be exploited to add resonances to homogeneous floors in order to change the perceived ground material: e.g., a floor made of *concrete* can be augmented so as to resonate like a *wooden* floor.

Notice that the implementation of the impact model currently being used is provided as a separate binary inside the directory `extra_binaries`. Make sure to copy it in your pd externals directory or alternatively in the directory containing the pd patches.

OSC version The OSC version consists in two patches organized in a *client-server* fashion (see Fig. 3.1): one patch for the sound **synthesis** part and one for the **control** part (respectively named `SOFT_IMPACT_OSCsrv.pd` and `SOFT_IMPACT_OSCcl.pd`).

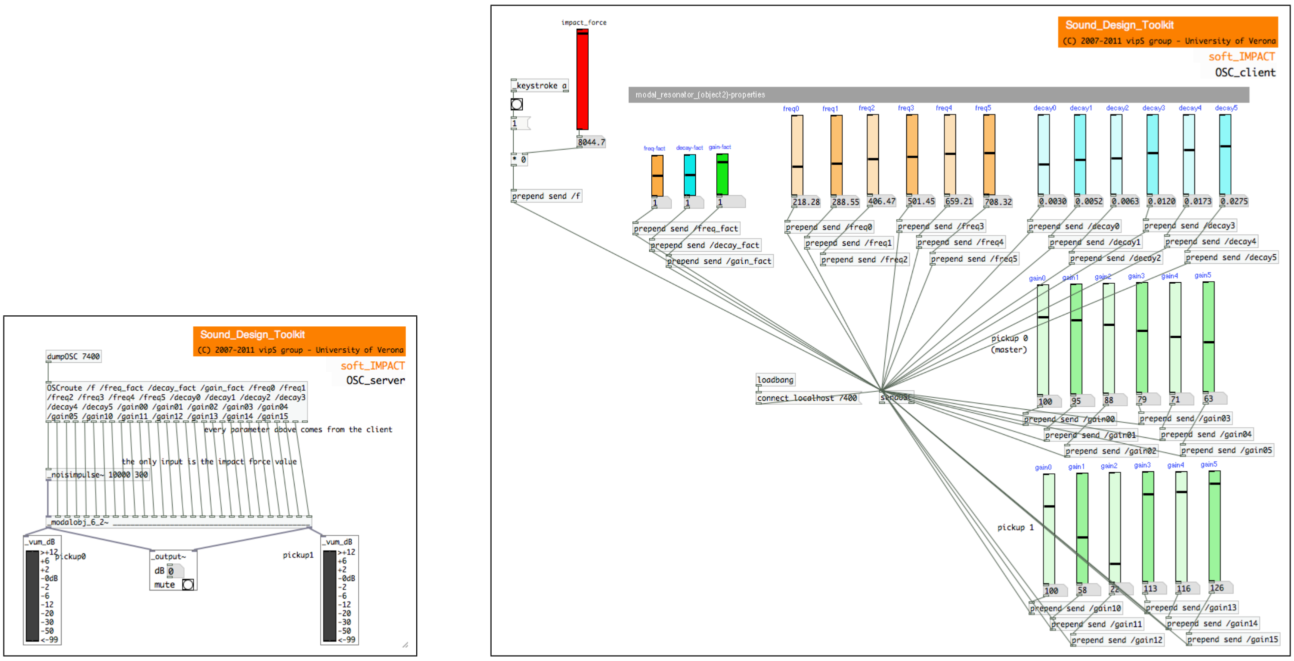
NIW shoes version The model is organized in a *client-server* fashion (see Fig. 3.2), making use of pd's internal message-passing functionalities. In detail, the server patch (`SOFT_IMPACT_srv.pd`) contains the synthesis engine, while several client patches are provided as sound presets (named `preset_SOFT_IMPACT_##_cl.pd`, where `##` stands for a two digit number). Start the server patch first and then one of the preset patches in order to correctly initialize the synthesis engine.

This version assumes that four data streams (coming from the four force sensors inserted into the shoes) are provided: each of them drives the amplitude of a different *noise burst* exciting one of four different impact models. The four impact models share common parameters for the resonators, thus allowing to simulate a uniform ground sound.

[3] Available at `http://puredata.info/downloads`

[4] Open Sound Control. More information at `http://opensoundcontrol.org/`

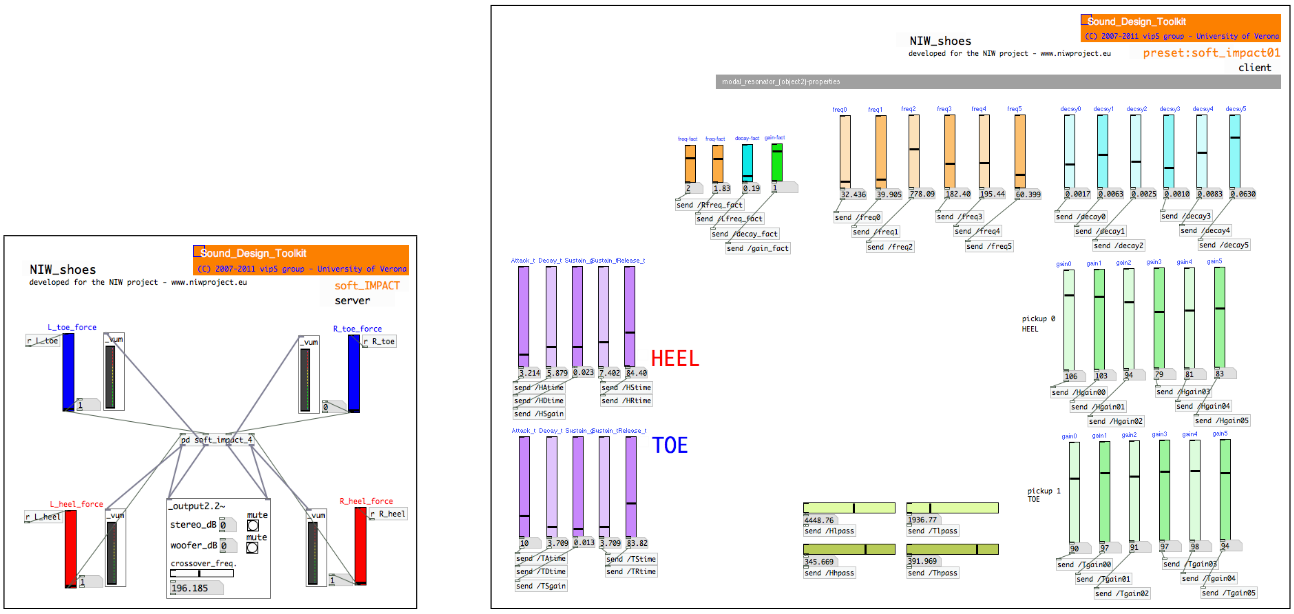
[5] `http://www.arduino.cc/`



(a) OSC server patch.

(b) OSC client patch.

Figure 3.1: OSC-based patches for the soft impact model.



(a) Server patch.

(b) Client patch (sound preset).

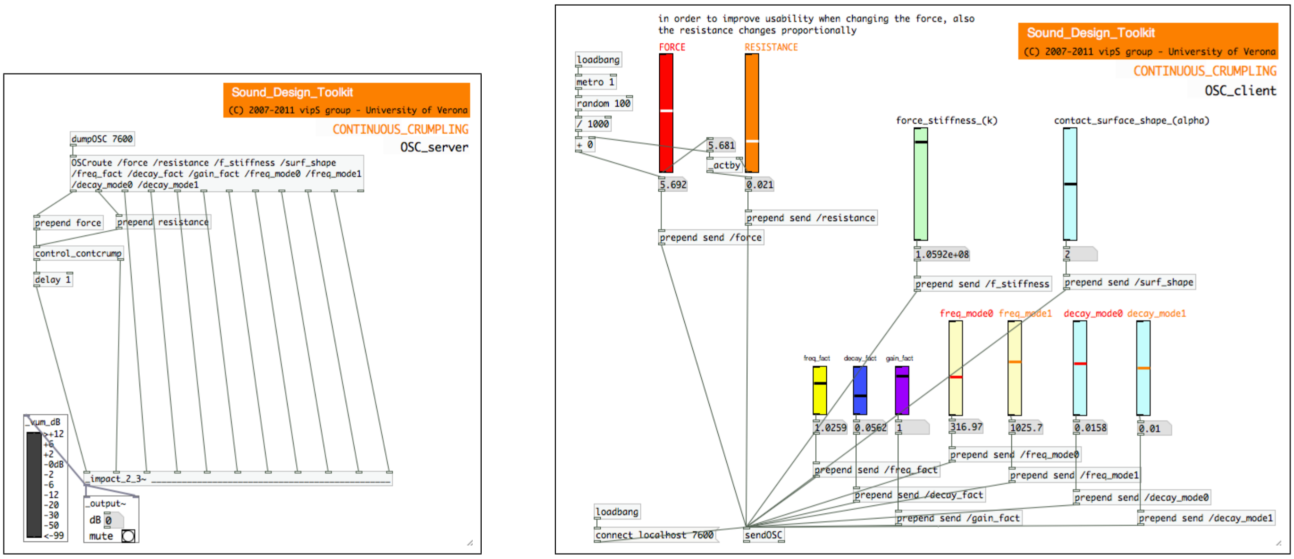
Figure 3.2: Patches developed for the NIW shoes and implementing the soft impact model.

3.2.2 Continuous crumpling

The *continuous crumpling* model is based on the impact model, on top of which a statistics of temporal impact events is superimposed. The model has already been successfully exploited to synthesize sounds and render the haptics of walking on aggregate grounds, such as *snow* and *gravel* [9, 50].

OSC version Two patches organized in a *client-server* fashion are provided (see Fig. 3.3): one patch for the sound **synthesis** part and one for the **control** part (respectively named CONTINUOUS-

_CRUMPLING_OSCsrv.pd and CONTINUOUS_CRUMPLING_OSCcl.pd).



(a) OSC server patch.

(b) OSC client patch.

Figure 3.3: OSC-based patches for the continuous crumpling model.

NIW shoes version The model is organized in a *client-server* fashion (see Fig. 3.4), making use of pd's internal message-passing functionalities. In detail, the server patch (CONTINUOUS_CRUMPLING_srv.pd) contains the synthesis engine, while several client patches are provided as sound presets (named **preset_*_##_cl.pd**, where * is the name of the preset, and ## stands for a two digit number). Start the server patch first and then one of the preset patches in order to correctly initialize the synthesis engine.

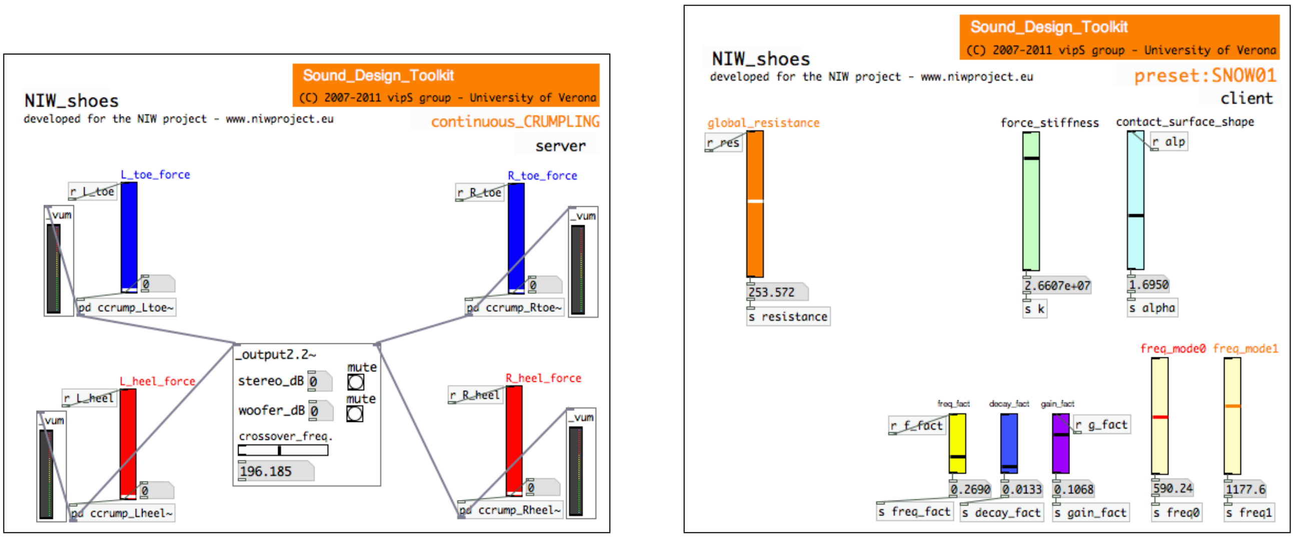
Again, this version assumes that four data streams (coming from the four force sensors inserted into the shoes) are provided, each driving the force parameter of a separate crumpling model. The four crumpling models share common parameters for the resonators, thus allowing to simulate a uniform ground sound.

In order to add a credible variation to the sound of the toe and the heel, the factors multiplying the modal frequencies of the resonators are internally transposed: in more detail, to the toe corresponds a higher pitch, while to the heel a lower one. Moreover, an additional small offset has been added to the frequency factors for the left and right shoes, in order to have a consistent yet slightly different sound corresponding to the two feet.

3.3 Physically informed sonic models

For reasons which have already been introduced in the beginning of this chapter, the strict use of physical models can be conveniently complemented by physically informed sonic models (PhiSM), proposed by Perry Cook [12].

This model simulates particle interactions by using a stochastic parameterization. This means that the different particles do not have to be modeled explicitly, but only the probability that particles will create some noise is simulated. For many particle systems, this phenomenon is well taken into account by using a simple Poisson distribution, where the sound probability is constant at each time step, giving rise to an exponential probability waiting time between events.



(a) Server patch.

(b) Client patch (sound preset).

Figure 3.4: Patches developed for the NIW shoes and implementing the continuous crumpling model.

3.4 Toward a generalized walking synthesizer

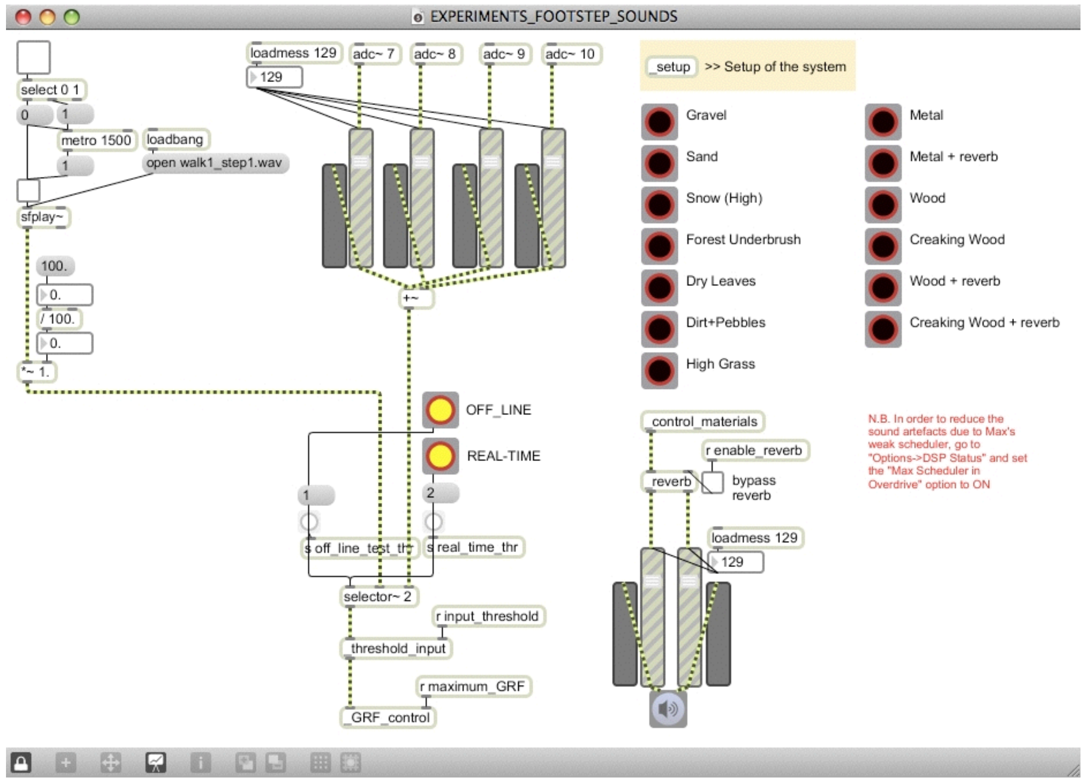
In order to investigate how the combination of parameters in the different basic models described in the previous section affects the perception of material, a generalized walking synthesizer has also been built. As solid surfaces, metal and wood have been implemented. In these materials, the impact model was used to simulate the act of walking, while the friction model (refer to Section 2.2) was used to simulate the sound of creaking wood. As aggregate surfaces, gravel, sand, snow, forest underbrush, dry leaves, dirt plus pebbles and high grass have been implemented. A footstep sounds is dependent on the kind of shoes the subject wears and obviously the kind of surface the user is walking on. We synthesized all the sounds assuming that the shoes hitting the floor had a solid soil. This aspect is more important in the simulation of solid floors rather than of the non homogeneous ones.

The algorithms to synthesize such sounds have been developed combining a spectral analysis of recordings from real footsteps with some ad-hoc manipulations of the control parameters of the different algorithms. The starting point has been the listening of the recordings of real footsteps sounds, in order to extrapolate the main features characterizing the sound of the footsteps on each surface, giving particular attention to those simply recognizable to a first listening. Indeed, the different components clearly noticeable in each sound, i.e, different sub-events that characterize the sound itself, have been took into consideration, with the aim of simulating independently them and their evolution in the time, and subsequently of combining them appropriately in order to construct the wanted global sound. The amplitude of the different components were also appropriately weighed, according to the same contribution present in the corresponding real sounds. Finally, a scaling factor for the subcomponents volumes gives to the whole sound an appropriate volume, in order to recreate a similar sound level which it would happen during a real footstep on each particular material. The sound synthesis algorithms were implemented in C++ as external libraries to the Max/MSP sound synthesis and multimedia real-time platform.

A screenshot of the final graphical user interface can be seen in Figure 3.5.

3.4.1 Solid surfaces

Wood and creaking wood The sound of the footsteps on wood has been synthesized by means of the impact model—refer to Section 2.1. In particular the GRF controls only the impact force parameter, while all the other parameters do not change their status.



N.B. In order to reduce the sound artefacts due to Max's weak scheduler, go to "Options->DSP Status" and set the "Max Scheduler in Overdrive" option to ON

Figure 3.5: Screenshot from the synthesizer simulating different surfaces.

Some creaking sounds have also been simulated, in order to realistically synthesize the typical sounds of footsteps on parquet floors. Such sounds have been generated by means of the friction model, controlling with random ramps the external rubbing force and pressure on rubber parameters. The ranges of variation and duration of such ramps have been set by means of random numbers, which are calculated each time the step is detected as finished. The simulation of creaking sounds enhances the realism of the sound of walking on wood, because their frequency, amplitude and duration change at every step. Such creaking sounds are also randomly generated, as it happens in real life.

Finally it is possible to enhance both the sounds of footsteps on wood and creaking wood with reverberation.

Metal The sound of the footsteps on metal has been synthesized by means of the impact model. In particular the GRF controls only the impact force parameter, while all the other parameters do not change their status.

Various kinds of metal can be synthesized thanks to the model. Moreover it is possible to enhance such sounds with reverberation.

3.4.2 Aggregate surfaces

Gravel The footsteps sound on various kinds of gravel were simulated by means of the combination of one, two or three PhISM models. The only limitation to such a combination concerns the computational load, which has to be taken into account especially in a real time context.

The sound of a footstep on gravel is composed by the contribution of the sounds of stones of different dimensions, which when colliding give rise to different random sounds with different features. It is possible to reproduce such a complex sound by setting appropriately the parameters of the models in order to simulate one, two or three kinds of distinct collisions between stones of the same dimension.

The system energy parameter of each model is controlled by the incoming input GRF. In this way a higher GRF creates a higher amplitude of the sound produced. In addition to this, the GRF is mapped to the system decay parameter, and this allows to simulate the degree of dispersion of the little tones from the foot.

Finally the number of colliding object parameter is calculated randomly each time the step is finished, while all the other parameters do not change their status.

Beach sand The sound of the footsteps on beach sand has been developed by means of a single PhISM model. The GRF is mapped to the sound decay parameter while all the other parameters do not change their status. The global volume for this sound is low as also in the reality such a sound is cushioned, and such a feature is held also for high values of the GRF controlling the system energy parameter.

Forest underbrush Three PhISM models working in parallel are used to synthesize the complex sound of footsteps on forest floor. The first model simulates the fall of the foot on a dirt floor, the second the trampling on the underbrush of the forest (e.g. leaves), and the third the breaking of little branches under the foot.

The volumes of the three models change at every step thanks to random calculations, so their balance varies continuously during the walk. In particular a further control on the volume of the third model allow to simulate the random breaking of the branches, which does not happen at every step neither with the same intensity.

The GRF controls the system decay parameter of the first and second model, while the detection of the end of the sound produced by the algorithms activates new random calculations controlling the number of colliding objects for the three models and the system decay parameter for the third model.

All the other parameters of the three models do not change their status once set.

Dry Leaves The footstep sound on dry leaves sound is a combination of granular sounds with long duration both at low and high frequencies, and noticeable random sounds with not very high density that gives to the whole sound a crunchy aspect. These three components have been reproduced with as many PhISMs with the same characteristics of density, duration, frequency and number of colliding objects. The first and second models simulate two layers of leaves at high and low frequencies respectively, while the third model implements the crunchy sounds.

The GRF controls the system decay parameter of the first and second model, while the detection of the end of the sound produced by the algorithms activates a new random calculation for the control of the number of colliding objects of the third model.

All the other parameters of the three models do not change their status once set.

Dirt plus pebbles Three PhISM models working in parallel are used to synthesize the sound of footsteps on a country road with dirt and some pebbles. The first model simulates the fall of the foot on a dirt floor, while the second and the third the trampling on some pebbles of two different kinds.

The volumes of the second and the third models change at every step by means of random calculations, so the balance of the contribution to the sound of the three models varies continuously during the walk.

The GRF controls the system decay parameter of the three model, while the detection of the end of the sound produced by the algorithms activates new random calculations controlling the number of colliding objects for the three models.

All the other parameters of the three models do not change their status once set.

High grass Three PhISM models working in parallel are used to synthesize the sound of footsteps on high grass (about 30 cm). The first and second models simulates the trampling on two different grass bundles, while the third adds some crunchy sounds. The detection of the end of the sound produced by the algorithms activates two new random calculations for the control of the number of colliding objects of the second model and the volume of the third model (not at every step the crunchy sounds are present).

Deep and low snow The footsteps sound on snow has been synthesized by means of one PhISM model and one crumpling model in order to simulate the two subcomponents of the sound produced when the foot drops into the snow and the snow breaks under the foot respectively. Deep and low snow have been developed thanks to different settings of the parameters of both the models.

The incoming input GRF controls both the system energy parameter of the PhISM model and the volume of the crumpling model. In this way higher the GRF and higher is the amplitude of the sound produced, as it happens in the reality when the foot drops into the snow with various intensity.

As concerns the PhISM model the GRF is mapped in the system decay parameter, and when the sound produced by the algorithms is detected as finished, a new random number is calculated to control the sound decay parameter. As regards the crumpling model the GRF is mapped into the range of variation of the impact force parameter, while when the sound produced by the algorithms is detected as finished, new random numbers are calculated to control the density of the crumpling effects, and the parameters resistance, contact surface (α) and decay of mode number 0.

All the other parameters for both the models do not change their status once set.

3.4.3 Controlling the synthesizer

The footsteps synthesizer described in the previous section has been designed to be controlled by an unique input parameter, that is the ground reaction force (GRF). The GRF has been normalized in order to be a time-varying continuous curve ranging on a scale [0,1]. To this purpose a setup control has been developed in order to detect a plausible maximum input value to be scaled to 1 (and all the values bigger than such a maximum are clipped to 1). Moreover, a threshold to eliminate any background noise was placed in order to set to 0 any incoming value under it. In this way the algorithms developed work independently from the system which the GRF is detected with (tiles and shoes augmented with sensors, acoustic waveform, or other apparati, etc).

At the moment all the developed algorithms have been extensively tested using as estimated GRF the envelope extracted from the acoustic waveform, as described in Deliverable 3.1. The synthesized sounds can be generated both in real time and off line, by extracting the envelope from real footsteps sounds captured by microphones and from a file with a recorded walk respectively.

The working of the algorithms is also based on the control of both the end of each step and the end of the footstep sound produced by the algorithms themselves. Such informations are extrapolated from the detected input GRF, thanks to a system of thresholds[6]. Each time that a step is finished a new combination of some parameters and some amplitudes, is calculated for the synthesis of the next step thanks to random numbers varying in appropriate ranges. Such a behaviour allows to increase noticeably the degree of realism of the proposed sounds as in real life the sound of each step is different from the previous. Finally the input GRF controls directly all or some of the parameters of the various algorithms, as well as the range of variation of the amplitudes of both the subcomponents and global sound. One of the challenges in implementing the sounds of different surfaces was to find the suitable combinations of parameters and their range of variations which provided a realistic simulation.

[6] For efficiency in computational load the end of the sound produced by the synthesis algorithms is set to 400 ms after the detected end of the step. Indeed on average the duration of the sound on the various materials does not last after such a temporal interval. In this way only the control on the end is performed.

4 Sounds from interaction with liquids

4.1 Introduction

Among the wide range of walking generated sounds, those involving the interaction with a liquid lying on the floor are of certain interest for the NIW project. This motivated a preliminary study concerning the nature and the modeling of the sounds arising from phenomena involving liquids.

The literature concerning noise emission due to liquid-solids and liquid-liquid interactions, including liquids in motion, addresses a number of phenomena ranging from the case of a single drop impact event (dripping), to more complex ones involving large liquid masses and complex evolutions in time. In general, large scale phenomena with complex time evolution are analyzed and modeled by looking at the whole phenomenon as the sum of a large number of elementary phenomena. One of the most important and acoustically relevant is the formation of air bubbles under the liquid surface, due to drops or solid objects impacting on a liquid surface [30, 34, 39]. In its simplest form, the acoustic emission is principally due to the initial impact and to the pulsation of the bubble. Depending on drop mass, impact angle, velocity, and other factors, the simple dripping event may turn into a variety of slightly different phenomena including liquid drop bouncing, spreading, and splashing (the formation of many secondary droplets at impact time), that have been accurately described and measured [43, 21]. Single-drop splashing is the simplest event which can be characterized by a structured time evolution in which the repetition of elementary events can be observed (principal drop impact followed by secondary droplet impacts). Other larger scale phenomena include sloshing, a term used in the literature to indicate the noise associated to a liquid in automotive fuel tanks which undergoes continuous shaking [1], and breaking waves [29].

The study of the acoustics of footsteps in water may be addressed at a first glance by focusing on the nature of splashing sounds. An exploratory investigation has been conducted in this first period of the project, having as principal objective the identification of a modeling scheme suitable for the generation of sounds related to impact events involving liquids. The first investigation focused on signal models which includes a minimal physically-based component. The second investigation focused on a scheme which is far more involved with the underlying physics, i.e. it addressed particle-based simulations of liquids dynamics adapted to splashing-like experiments, and in particular, the mapping between particles dynamics and the triggering of acoustic events.

4.2 Splash sound synthesis based on dripping model and texture sounds

Sound emission due to liquids in motion is known to be due to a restricted number of elementary phenomena, the most important one being the radial oscillation of air bubbles trapped below the surface of a liquid volume [22, 47]. The formation of radially oscillating bubbles under the surface of a liquid volume is often modeled assuming that the impulse response of the single bubble is well represented by a damped sinusoid with time varying frequency, given by Minnaert's formula. The law that describes the impulse response of such an event represents the radiated acoustic pressure as a damped sinusoid with rising frequency [47]:

$$p(t) = a \sin(2\pi f(t)t)e^{-dt} \tag{4.1}$$

where $f(t)$ is the time-varying pulsation frequency, d is the damping factor, a is the amplitude, and t is time. The simulation of a single bubble relies on the following formulas, reported in [47]: initial frequency is $f_0 = 3/r$ (where r is the bubble radius), the damping factor is $d = 0.043f_0 + 0.0014f_0^{3/2}$, and the instantaneous frequency is $f(t) = f_0(1+\sigma t)$ (where σ is the parameter which governs the slope of the frequency). The initial bubble radius is determined by a number of factors, i.e. the presence of a resting liquid level which permits the formation of the bubble under the surface, and the mass of the impacting solid or liquid. The frequency slope is related to the depth of bubble formation under the liquid surface.

One of the most common events involving the formation of radially oscillating bubbles under the surface of a liquid volume is dripping, the falling of a drop or an object into a quiescent liquid. In the simplest case, when the initial impact sound is neglected and when there is no crown formation following the initial impact, dripping can be represented by a single bubble event. However, the more the initial impact sound is perceivable and the cavity of the bubble becomes larger (e.g., for large impacting mass), the less the simple single bubble sound model becomes adequate to represent the dripping event. This is even more evident when large objects or drops falling into a resting liquid generate many secondary bubbles and droplets events due to the mass displaced by the principal impact event (splashing).

Table 4.1 shows the link between the two principal parameters used for the dripping model and their acoustic counterparts.

Sym.	Physical Description	Phenomenological Description
r	bubble radius	affects the initial pitch f_0
σ	density of the liquid	affects the speed of frequency rise

Table 4.1: A phenomenological guide to the dripping model parameters.

The more the cavity of the bubble becomes larger, the more the simple single bubble sound model becomes inadequate to represent the dripping event. This is most probably due to the fact that large objects or drops falling into a resting liquid generate a large cavity (single resonance) as well as many secondary bubbles and droplets events due to the mass displaced by the principal impact event. As a preliminary attempt to reproduce the temporal structure of a splashing sound, we choose to design this event as a sequence of three distinct events: 1. a short samples noisy impact sound, 2. a bubble sound modeled as detailed above, and 3. a secondary droplets event texture. The model has been implemented using sampled waveforms to reproduce the initial impact sound and the final sound due to droplets formation, whereas the principal bubble formation is based on the single bubble model.

4.3 Splash sound synthesis based on particles models

A second modeling approach, relying more strictly on physical motivations, was used to further investigate the control of sound emission involved in liquid phenomena. The rationale behind this choice is that the exploitation of the physics underlying a natural process may be in many cases the more effective way to understand, to control or to represent the process itself (examples can be found in the studies on phonation, predictive control of industrial plants, or animation through computer graphics). Physically-based process modeling has also proven to be an effective mean for real-time synthesis of sounds in many fields, including musical performance, virtual and augmented reality, HCI, interactive simulations of physical processes, computer graphics and animation[48].

In this investigation we focused on the generation of the sound produced by a liquid in motion, in the particular case in which the simulation is based on physically-based numerical models of the underlying fluid dynamics. The topic is particularly interesting for applications based on computer graphics and animation, where realistically animated fluids can substantially improve the perceived quality of rendering and interaction. However, even though these models are highly effective for the computation of the parameters needed to graphically represent the motion of a fluid, (usually with a frame rate of 30 fps), their computational load at audio sampling frequencies becomes prohibitive. Thus, we decided to adopt a hybrid solution, in which the particles motion is used to control, by means of opportune mappings, the audio synthesis algorithms used for the elementary sounds, such as the ones due to the formation of bubbles under the liquid surface or the impacts emphasized by resonant cavities, responsible for the acoustic emission in liquids. In the design of the algorithms, a “cartoonification” approach was followed, with the intention of emphasizing the prototypical characteristics of the events, instead of targeting the acoustic realism typical of the audio sampling techniques[45]. However, all solutions were supported by an an accurate investigation on the physics underlying the events under study. This approach is somehow related to the PhISEM approach by P. Cook [12], in which “off-line” particle interaction simulations are performed to derive statistical collision distributions and to drive the synthesis of sounds produced by pools of interacting objects (as in maraca, sleigh bell, whistles), including water drops sounds.

This study will consider an experimental configuration representing the falling of a liquid volume into an underlying rigid container. This type of setup is typical of a class of benchmark tests known in the literature as “breaking dam”, in which the dam walls holding the liquid suddenly break apart, provoking a flood wave to fill the basin (see Figure 4.1 and the video available from `www.niwproject.eu`).

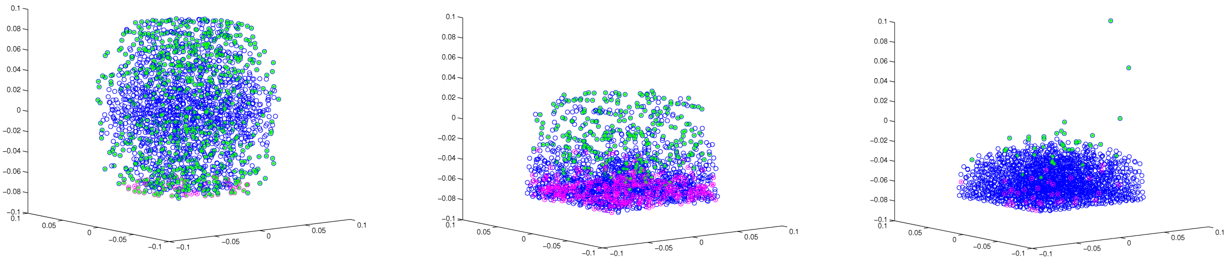


Figure 4.1: Three frames of the particles system representing the falling of a liquid into an empty container. a) impact of falling liquid with the empty container, b) container is filling up, and c) toward resting conditions. Different colors represent given properties or events of interest: particles with null velocity along the vertical axis are depicted in blue, falling particles are in green, and particles for which an abrupt change in the vertical velocity component (collisions) are in pink.

As a general overview, the framework consists of a numerical model of fluid dynamics based on smoothed particles hydrodynamics (SPH), an analysis model for classification of instantaneous particles configuration, and a minimal set of low level acoustic models representing basic events responsible for acoustic emission in liquids. In this study, we considered sounds originating from the formation of single pulsating cavities (bubbles) under the liquid surface (such as in dripping, or pouring), and from surface impacts (such as those occurring between two liquid volumes or between a solid and a liquid surface). At each frame, the proposed sonification scheme involves the update of the particles system through a smoothed particles hydrodynamics (SPH) solver, the analysis of particles configuration, and the mapping of events, which are known to lead to sound emission, into low level audio events rendered through opportune synthesis models.

4.3.1 Method

We focus here on a particle-based fluid simulation method known as smoothed particle hydrodynamics (SPH), which has interested many researchers and practitioners in the field of computer graphics and animations, due to the realistic nature of the animations of gas and liquids [32, 17]. In SPH, fluids are modeled with particles representing small fluid volumes. Each particle is described by a set of physical quantities (i.e. position, velocity, acceleration, pressure, and density) which are updated at each simulation step based on the interaction with neighbor particles. This interaction is ruled by the Navier-Stokes equations for the conservation of mass and momentum. Moreover, external forces such as the ones due to gravity or collisions with solids objects, can be accounted for in the simulation.

To address in a rigorous way the reconstruction of the acoustic field due to a system of particles in motion, one should address the calculation at audio rate of pressure wave radiation at liquid boundaries. However, this would be impractical due to computational complexity, because of the high temporal and spatial resolution requirements that this would imply. Our approach is to adopt commonly used computation rates for particles update (usually 10 to 30 frames per second) and to detect events for which acoustic emission is predictable. When such an event is detected, a corresponding acoustic event is triggered. A similar approach is found in the literature to treat the rendering of the vortex sound which accompanies air volumes in motion [51].

A preliminary step of this study concerned the selection of a minimal set of low level models representing basic events responsible for acoustic emission in liquids. In particular, the investigation focused on sounds originating from the formation of single resonating cavities (bubbles) under the liquid surface (such as in dripping, or pouring), and from surface impacts (such as those occurring between two liquid volumes or between a solid and a liquid surface) [21, 47].

In the following, we will first discuss the models used for the synthesis of the elementary acoustic events, then the analysis of the particles system and the reconstruction of the sound due to the liquid evolution will be illustrated.

4.3.2 Low level models for the synthesis of elementary acoustic events

The typical sound originating from a liquid drop falling into a liquid at rest is due to the formation, below the surface, of air bubbles with pulsating radius. The pulsation frequency is in the audible range, and rises with time, as already discussed in section 4.2.
At the time at which the falling drop or solid impacts the surface, a brief impulsive noise is also generated. This noise is characterized by resonances well visible in the spectrum due to the fact that the impact impulse excites cavities that are originated by air trapped between the impacting surfaces. Due to the non-rigid nature of the volumes involved in the impact, the resonances exhibit a time-varying nature and typically it is possible to observe, in the spectrograms of experimental recordings, non stationary "formant" patterns. Consequently, we decided to model this noise source by means of a subtractive synthesis in which one or two second order IIR filters are excited by a short impulse at the instant of impact. A similar model, based on a LPC analysis/synthesis scheme, was used in [37] to represent the sound of clapping hands, which has very similar perceptual characteristics to the impact noise under discussion. Figure 4.2 shows the spectrograms of a bubble event and of an impact event.

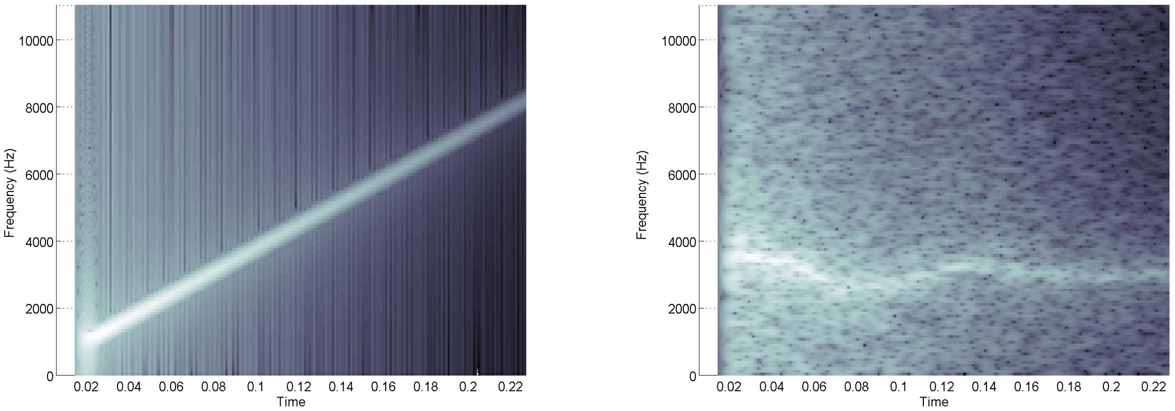


Figure 4.2: Spectrograms of the basic sound events employed: bubble (on the left) and impact (on the right)

4.3.3 Analysis and processing of the particles motion

At each frame, the particles analysis and sound generation process involves three main tasks:

1. the state of particles is updated through the SPH solver
2. the new particles configuration is analyzed, and events which are known to lead to sound emission are detected and classified
3. the detected events are mapped into low level audio event triggers with opportune synthesis parameters

A detailed description of the technique to solve the equations for the particles motion via the SPH method can be easily found in the literature (e.g., [17]). We will discuss here in detail the other two points.

The analysis of the instantaneous fluid configuration has a central role in the whole process. To this purpose, a set of relevant configurations and events, commonly observed in experimental situations involving liquids in motion, was identified. Among the configurations that were considered interesting, two are worth noting: a) the impact of a falling liquid volume with an empty container, and b) the impact of liquid volumes with the surface of a resting liquid in the container. In the first case, there is no possibility for the bubble formation phenomenon to occur, and only the impact sound due to the collision of the liquid volume with the solid surface may arise (Figure 1, panel a)). In the second case, together with the sound from liquid-liquid impact, sound from the formation of cavities under the liquid surface (bubbles) is also generated (Figure 1, panels b) and c)). The system evolution proceeds with particles falling until a terminal quiet state is reached, with some of the particles possibly bouncing a few times.

During the evolution, each particle is monitored and a list of neighbor particles is generated. This allows to obtain interesting information, such as the emergence of clusters of particles proceeding tight together (small volumes), isolated particles (drops), or the formation of noticeable configuration such as liquid-solid and liquid-air interfaces. In the experiment discussed here, during the particles analysis we rely on motion information (particles velocity, position, acceleration) and lists of neighbor particles. Based on these parameters, the following features are evaluated:

- the presence of resting liquid in the container. When the container is not empty, an estimation of the level h of liquid fallen up to the current frame is produced
- the presence of particles undergoing a collision (this is based on the acceleration)
- the identification of clusters of particles and of singleton particles in air
- formation of boundary configurations, i.e. horizontal liquid-air interfaces, at the bottom or top of a liquid volume

Once completed, the analysis of the particles configuration for the present frame is followed by the generation of control signals for the triggering of audio synthesis. The mapping between the states of the liquid description and the audio events was designed by searching for coherent relations between the parameters of particles motion and the acoustic parameters of the audio algorithms. The principal rules used to this aim are the following:

- the generation of sounds originated from the formation of air bubbles is dependent on the presence of resting liquid in the container, and is due only if $h > 0$
- for each collision detected, an impact sound is triggered, and a bubble sound is if $h > 0$. The trigger instants are modeled by a statistical uniform distribution within the audio frame corresponding to the analyzed video frame
- when clusters of collisions are detected, a bubble sound and an impact sound event are triggered, with resonance frequency inversely proportional to the cluster dimension
- for increasing values of h (increasing levels of liquid at rest in the container), an increment of the parameters controlling the bubble dimension and the bubble frequency slope (bubble collapse) is allowed.

In Figure 4.3, the spectrogram of the sound synthesis resulting from the adoption of the criteria described above is shown. It is possible to observe that when the first collisions occur (at around 0.3 s), only audio impact events are generated, without any bubble sound. After a few frames (at around 0.4 s), when the level of liquid in the container has reached a given threshold, bubble sound events,

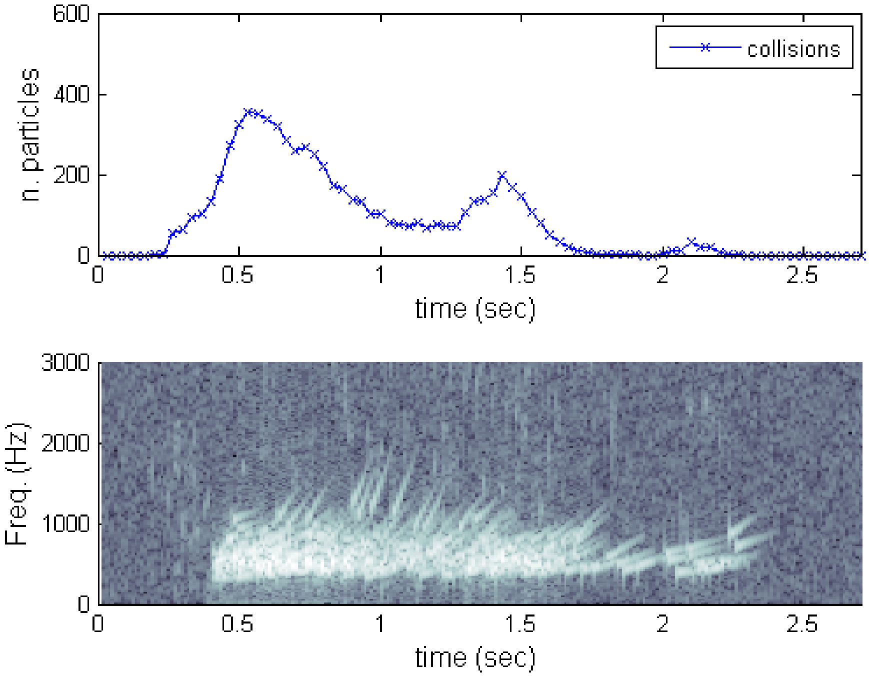


Figure 4.3: Result of the triggering of acoustic events based on the analysis of the particles configuration. Upper Figure shows the number of particles undergoing a collision. Lower Figure shows the spectrogram of the sound resulting from the combination of impact events and of bubble formation events.

recognizable from the characteristic rising pitch, are also visible in the spectrogram. It can also be observed how the slope of the bubble pitch is lower at the beginning, for low values of h, and increases as the container fills up.

4.4 Experimental results and discussion

Numerical simulations of the illustrated splashing event were conducted, and the analysis and audio synthesis steps were assessed. In general, the resulting audio-visual animations were judged satisfactory and perceptually in agreement with the average everyday The mean values of the statistical distributions of bubbles control parameters (bubble radius and frequency slope) and of impact resonances, were tuned empirically until the audiovisual results produced were perceptually in agreement with the average everyday experience of the experiment under study. Based on informal listening tests, we can say that the audio synthesis, even if isolated from the visual rendering, is well recognized and evokes a physical event involving the pouring of liquid into a bowl. The quality of the audio rendering is far from being as realistic as a sampled sound, however the perceptual information needed to convey the sensation of the source event is contained in the sound. The audiovisual result, in which the synchronization between the particles motion and the timbre variations of the acoustic events is clearly perceived, is coherent with our perception of the phenomenon.

4.5 Ongoing and future work

Future work will possibly be directed toward the application of particle-based liquids simulation and sound generation to virtual and augmented reality scenarios strictly connected to the analysis/synthesis of walking gestures. This will require to define the geometry and the dynamics of a splashing event representing a solid flat body hitting first the surface of a liquid volume contained in a recipient of finite dimensions, generating a typical splash with secondary droplets formation, and finally hitting his bottom. This experimental setup, coupled with sound synthesis models of both the splash event and impact between two solids, would effectively represent a wide set of acoustic events to be reproduced when simulating walking in presence of a liquid lying on the ground.

5 Parametric strategies for walking sound synthesis

In parallel with the synthesis of walking sounds obtained using physically-based models, alternative approaches have been attempted by the consortium in the case when there was reasonable expectation to come up with convincing, low-latency sonic interaction. This activity has been started early on this August (i.e., month 11). In consequence of this relatively late start the proposed results, although already promising, are only partially systematic and quite preliminary.

Parametric sound synthesis has already been proposed for the synthesis of walking [18]. UNIVR has engaged a graduate student (Fabio Morreale) in charge of implementing a method for the parametric synthesis of walking sounds, that proved effective in the simulation of floors exposing both homogeneous and aggregate surfaces. This method inherits and introduces some novelties to an existing analysis-and-synthesis technique, that has been proposed by Cook for the generation of walking and, later, hand clapping sounds [13, 37]. Specifically, some changes of the method have been applied in an effort to keep the control of the synthesis process intuitive for the sonic interaction designer.

5.1 Enveloping filtered noise

The idea, indeed quite simple, is that of starting from the knowledge contained in a pre-recorded set of walking sounds. This knowledge is used to weight band-filtered noise through specific temporal envelopes: first, by tuning the center frequency and bandwidth of a series of simple (second-order) resonant filters; then, by figuring out enveloping functions by means of a simple envelope follower. By inheriting walking sounds across different floor materials from the free online database `www.freesound.org`, a significant palette of acoustic floor properties has been prototyped: snow, leaves, wood, and metal. The reader is referred to the sound samples available from `www.niwproject.eu`.

Figure 5.1 shows the method in more detail, as well as the procedure behind.

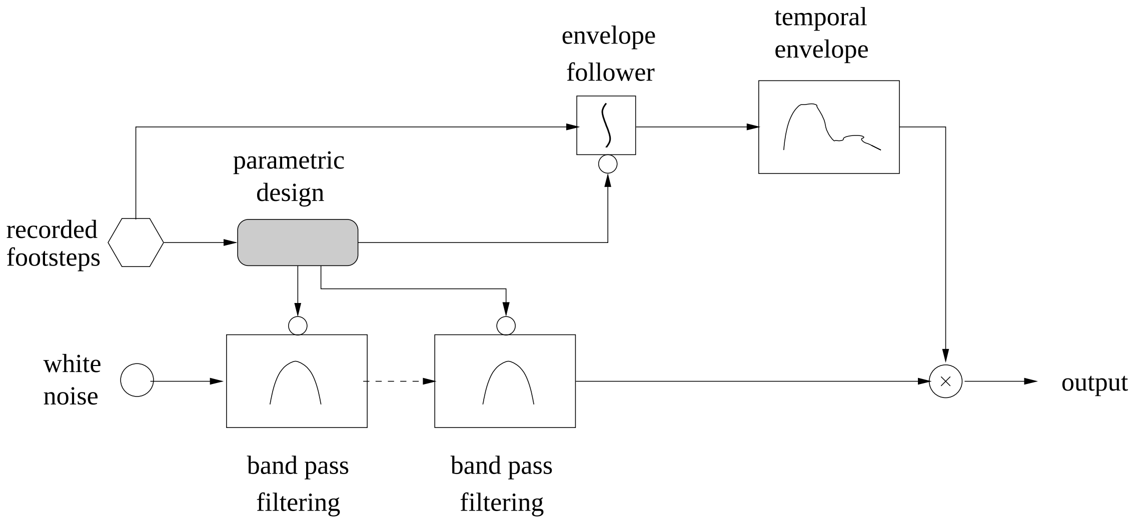


Figure 5.1: Sketch of the method based on parametric synthesis of footstep sounds.

As we can see from the figure, two basic steps are needed along this procedure:

- once the acoustic properties of a particular foot-floor interaction have been identified from a pre-recorded set of footstep sounds, parameters of center frequency and bandwidth are determined. At present this identification is made by subjective considerations—the identification of a formal procedure is left to future research. Taken together, these couples of parameters form the sonic characteristics of one or more digital filters which are in charge of transforming white noise into colored sounds for that interaction;
- the same recordings are used also to determine one or more "prototype" envelopes (currently one in the case of aggregate materials, and two, i.e. for the toe and heel positions, in the case of solid materials) through the following digital envelope follower [13]:

$$e(n) = 1 - b(n)|x(n)| + b(n)e(n-1) \tag{5.1}$$

in which

$$b(n) = \begin{cases} b_{up} & \text{if } |x(n)| > e(n-1) \\ b_{down} & \text{otherwise} \end{cases} .$$

Such prototypes are then used to shape the filtered noise across time, by giving it the temporal features that apply to the specific foot-floor interaction. The envelopes are then stored in the memory of the computer, in the form of arrays whose elements are read at every temporal step n during the synthesis of a single step sound.

At present, the samples that can be found on the web site are the result of a subjective choice of the filter parameters based on a visual inspection of the spectrograms of pre-recorded sounds, accounting for a specific type of ground. These parameters have been set as in Table 5.1:

Table 5.1: Second-order filter parameter values for different ground materials

Material	Filters	Bandwidth (Q)	Center frequency	Gain
Snow	IIR BP	50	400 Hz	1
	IIR BP	700	660 Hz	0.4
Dry leaves	IIR BP	50	100 Hz	1
	IIR BP	500	850 Hz	0.5
	IIR BP	5000	6000 Hz	0.33
Metal (heel)	IIR BP	17	220 Hz	1
	IIR BP	500	220 Hz	1
Metal (toe)	IIR BP	12	250 Hz	1
	IIR BP	20	250 Hz	1
	IIR BP	200	400 Hz	0.24
	IIR BP	500	400 Hz	0.24
Wood (heel)	IIR BP	250	250 Hz	1
	IIR BP	250	250 Hz	1
	IIR BP	180	660 Hz	0.01
	IIR BP	550	660 Hz	0.01
Wood (toe)	IIR BP	55	130 Hz	1
	IIR BP	250	130 Hz	1
	IIR BP	200	610 Hz	0.16
	IIR BP	500	610 Hz	0.16

Furthermore, the envelope functions have been obtained after temporal averaging of the same pre-recorded sounds. The coefficients b_{up} and b_{down} of the envelope follower (5.1) have been respectively set to 0.85 e 0.9975. For instance, the enveloping signal for the case of the metal floor interacting with the heel is plotted in Figure 5.2, while being superimposed to the recorded (above) and synthesized (below) sound.

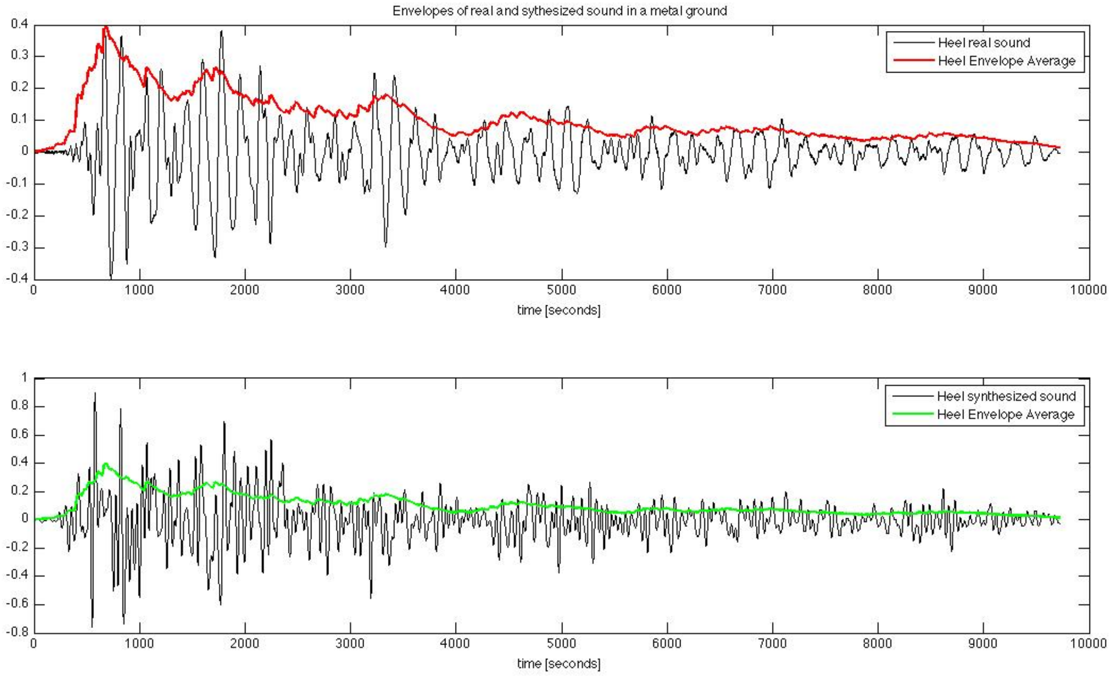


Figure 5.2: Enveloping signal for the heel/metal interaction case. Superimposed to the recorded sound (above). Superimposed to the synthesized sound (below).

Although such sounds need to be subjectively validated, their already acceptable quality is *per se* an indication of the affordability of the design process from the part of the interaction designer, who needs not to be an expert in sound processing: the issue of accessibility of the model to sound designers is of primary importance in the context of this research project.

Figure 5.3 compares the signal recorded when the ground is covered with snow (above) with its synthetic reconstruction obtained using the proposed method (below).

5.2 Ongoing work

Key in this process are ways in which the pre-recorded sounds are analyzed, then used to model the noise filter and the envelope functions under different interaction conditions. Currently, we envision possibilities to include the proposed model in a closed-loop interaction paradigm that instantaneously responds to the foot gesture during walking. As of today, we are starting tests in which the temporal envelopes are slightly randomized, then read at a velocity depending on the expected strength and duration of the Ground Reaction Force function. More precisely:

1. an averaged enveloping function $\overline{I}$ is obtained by averaging N envelopes $I_1, \ldots, I_N$ obtained by recorded sounds through the characteristic (5.1). The corresponding standard deviation σ_I of the same envelopes is obtained as well, using formulas from standard statistics;

2. when a walking event is detected, an envelope I is created sample-by-sample on the fly, by superimposing at every temporal step n a random trajectory to the average envelope $\overline{I}$:

$$I(n) = W_{T_D}\left\{\overline{I}(nT_D) + \alpha(n)\sigma_I(nT_D)\right\} \qquad (5.2)$$

The constant T_D is used in (5.2) to shrink the average envelope and its standard deviation according to the expected duration of the Ground Reaction Force, in a way that the higher this force, the shorter

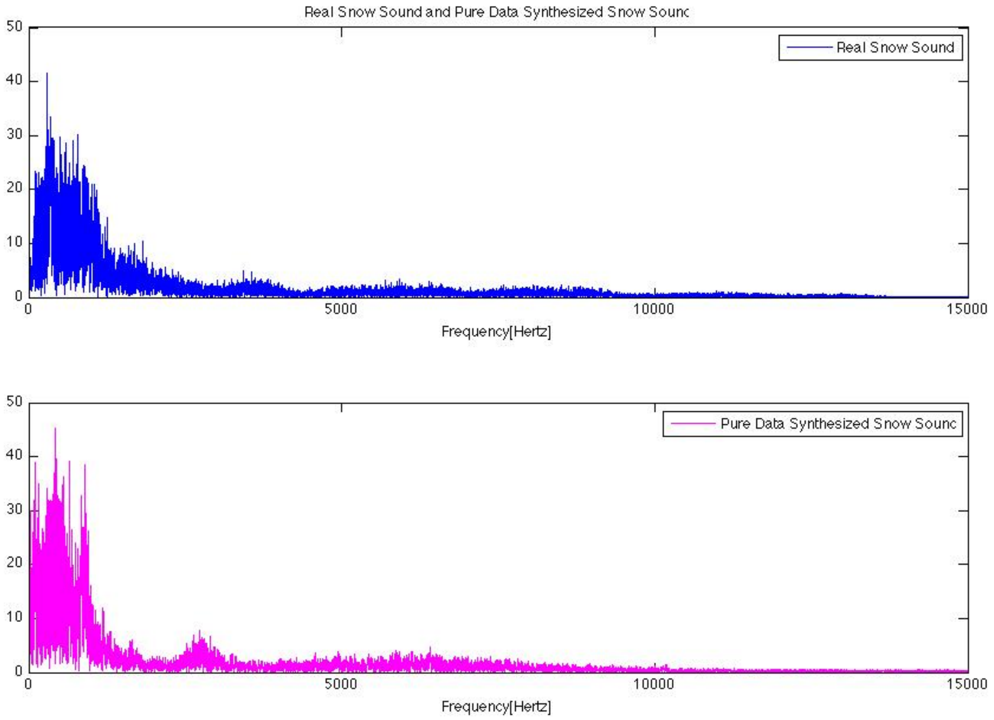


Figure 5.3: Sound signals for the foot/snow interaction case. (above) Recorded sound. (below) Synthesized sound.

the envelope is. For this reason it is $T_D \geq 1$. The temporal distortion of $\overline{I}$ and σ_I is obtained by simply interpolating between the two values of such signals that are closest to nT_D. This practice is common in real-time sound processing [44]. For instance, the interpolated average envelope at nT_D will be equal to:

$$\overline{I}(nT_D) = \overline{I}(\text{floor}[nT_D]) + \frac{\overline{I}(\text{ceil}[nT_D]) - \overline{I}(\text{floor}[nT_D])}{\text{ceil}[nT_D] - \text{floor}[nT_D]}\,(nT_D - \text{floor}[nT_D]), \tag{5.3}$$

the same interpolation being applied to the standard deviation σ_I.

In parallel with temporal shrinking, higher forces elicit larger envelopes: for this reason, we weight every instance of the envelope by a value W_{T_D} which, again, ultimately can be made depending on the Ground Reaction Force by means of the shrinking factor T_D. At the moment we are managing to set the value of W_{T_D} in a way to equalize the average energy of the envelope:

$$\sum_n \{W_{T_{D_1}}\, \overline{I}(nT_{D_1})\}^2 = \sum_n \{W_{T_{D_2}}\, \overline{I}(nT_{D_2})\}^2 \tag{5.4}$$

for any couple (T_{D1}, T_{D2}).

In order to have a tightly coupled closed-loop interaction, only the first few (3 or 4) milliseconds of a walking event can be spent for estimating the Ground Reaction Force function. Similarly to what we have done in the case of the real-time synthesis of physics-based sounds, we use the early attack of the signal coming from the FSR's on board of the augmented shoes through the Arduino board: the steeper this attack, the larger the expected force exerted by the foot into the ground.

Further ongoing activity on this theme deals with the recording at foot level of walking sounds, which will replace the currently used pre-recorded sets downloaded by freesound. Overall, results from this activity will be delivered as part of the planned project dissemination tasks.

6 Augmentation of walking activities with the mobile phone[1]

While figuring out cheap and simple ways to design interfaces providing augmented feedback to walkers based on their gestures, UNIVR wanted to test if it was possible to embed the whole computation and machine interaction level into the most compact and affordable computing and communication device available today: the mobile phone. The ultimate goal of this experience, hence, was to try to instantaneously augment a walking activity (particularly locomotion and shoe scraping) by means of auditory and haptic cues provided by a cellular phone capable of capturing the gestures of a walker.

Unfortunately, as we will see this experience was prematurely stopped, in consequence of the interruption of the research contract with the post-doc student who was in charge of following this activity. Nevertheless, it is instructive to document it carefully until the point at which it ceased.

Nowadays smartphones are provided with several sensing and actuation devices. Open software platforms on board of these devices allow third parties and independent programmers to access to their resources and process the data recorded from them. For our experience we have chosen a Nokia N95 Smartphone (equipped with firmware V21.0.016, OS 9.2, SDK S60 3rd FP1) , whose features and specifications can be found at `http://www.forum.nokia.com/devices/N95`. Its two CPU's (ARM 11 working at 332 MHz) and 3D onboard graphic accelerator allow to make fairly sophisticated digital signal processing. Though, this model does not have a separate DAC/DSP chip. Its computational power cannot be compared to that of a modern PC, but benchmarks (`http://www.glbenchmark.com/result.jsp`) and porting of games (such as Quake, `http://www.youtube.com/watch?v=JqwoYNYE4_w`) have been performed, suggesting that the N95 can be used to perform audio processing without big trouble.

The advantages of this platform are manifold: powerful computation thanks to its dual processor, and especially an open programming environment based on the Symbian operating system that allows to run C++ programs[2]. Besides these software resources, this smartphone includes a 3D accelerometer as well as a vibratory device and a loudspeaker. Figure 6.1 shows the N95 and the Cartesian axes adopted in this experiment.

The original research plan included the following tasks:

1. understanding the platform: its GUI, API's, CPU-featured DSP/audio interface, and communication with the accelerometer;

2. acquiring the data from the accelerometer;

3. designing a fast analysis procedure for these data;

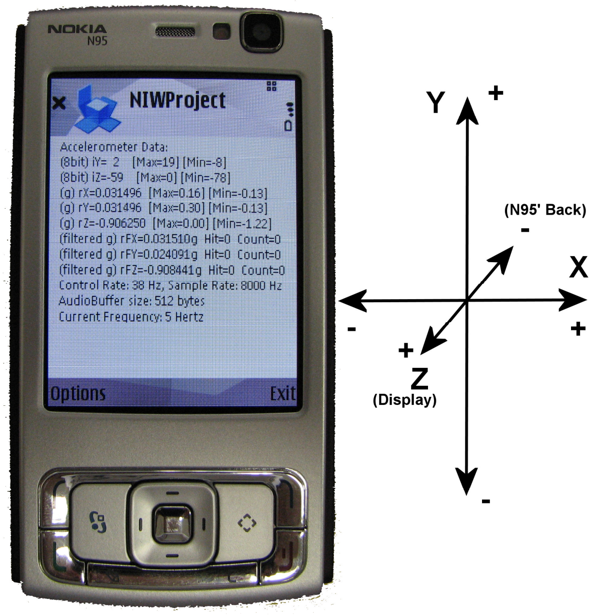
4. importing sound synthesis models in the platform, especially those available in the SDT;

5. experiencing with sonification paradigms for augmented walking and foot scraping, displayed by means of the mobile phone loudspeaker;

6. (similarly to the previous task) experiencing with tactile feedback paradigms displayed using the vibratory device.

[1] Currently interrupted.

[2] At the time when this experience started the Apple iPhone, now extensively hacked in many ways, did not provide the same accessibility at software level.



```
Accelerometer Data:
(8bit) iY= 2   [Max=19] [Min=-8]
(8bit) iZ=-59   [Max=0] [Min=-78]
(g) rX=0.031496  [Max=0.16] [Min=-0.13]
(g) rY=0.031496  [Max=0.30] [Min=-0.13]
(g) rZ=-0.906250  [Max=0.00] [Min=-1.22]
(filtered g) rFX=0.031510g  Hit=0  Count=0
(filtered g) rFY=0.024091g  Hit=0  Count=0
(filtered g) rFZ=-0.908441g  Hit=0  Count=0
Control Rate: 38 Hz, Sample Rate: 8000 Hz
AudioBuffer size: 512 bytes
Current Frequency: 5 Hertz
```

Figure 6.1: The N95 and the Cartesian axes adopted in this experiment.

6.1 Task 1: Understanding the platform

The Symbian OS is specialized for use on portable devices. A study on the API's has been made, to understand how to operate on this platform. As they are quite similar to those found in Microsoft Windows, understanding them did not pose peculiar problems. In particular, they allow for the control of the audio DSP. The GUI, intuitive and well designed, was not difficult to understand as well.

The control software for the accelerometer lies on a separate, proprietary package. This package provides several interface functions, which make the access to the accelerometer quite easy. Furthermore, Nokia freely distributes a comprehensive set of application examples that can be used to speed up the development of software code for the N95. Starting from these examples we created code for testing the components of interest (audio, accelerometers, GUI's, and so on).

6.2 Tasks 2 and 3: Acquisition and Data Analysis

Based on the previous tests, an application was developed capable of collecting data from the accelerometer, then saving them in the solid state disk of the mobile phone in the form of CSV files also containing precise information on the time detection of the same data. These files could be easily copied into the memory of a host PC for off-line analysis (for instance using Matlab).

At this point, some preliminary tests were performed by attaching the smartphone to the heel (`http://www.niwproject.eu/`, first group of pictures), and then recording data during a walk and when shoes were scraped over a homogeneous floor. Among other things, this task needed to resolve the association between the recorded data and the position of the foot. For this purpose, a video camera was used to record the experiments whose outcome was synchronized with the acceleration data.

Figures 6.2 and 6.3 show plots, respectively in the case of walking and foot scraping, obtained from these data after importing them into Matlab.

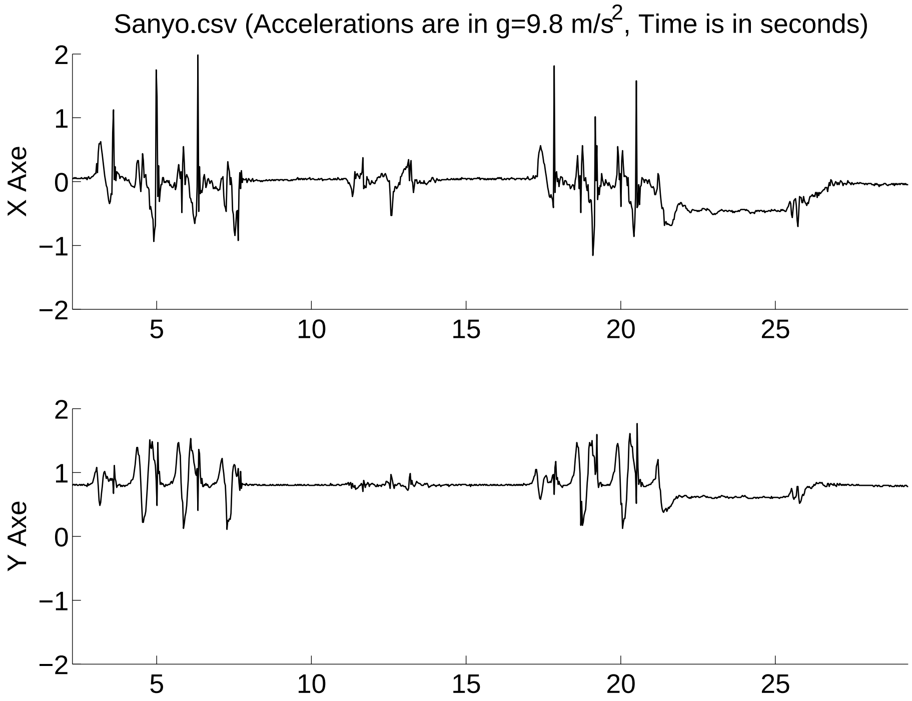


Figure 6.2: Example of data from the accelerometer, analyzed with Matlab (walk).

Figure 6.2 shows the recorded acceleration respectively along the horizontal (x) and vertical (y) axis. The figure can be logically divided into three parts: from 0s to 10s, from 10s to 15s and from 15s to 25s. i) The first part contains the acceleration values of the first step (from the beginning to the end of a pace). The mobile phone is attached to the leg that performs the pace. ii) The second part contains the acceleration data for the second step. This time the mobile phone is not attached to the leg performing the pace, hence the data are not relevant. iii) The last part is a "freezed" step., i.e., the pace has been interrupted and the leg stops in a steady position. The corresponding acceleration produces a negative, almost flat line visible in the graph that lasts until second 25. Finally, the foot relaxes over the ground.

Both interactions have some aspects in common. The simple visual examination of the plots shows that the acceleration is larger in the horizontal axis (i.e. along the direction of movement), not in the vertical axis as one could expect. The direction of movement, then, probably carries the most important information on walking. Correlation with the data belonging to the other directions of movement should have been performed at this point. This did not happen for the reasons mentioned in the beginning of this section. For the same reason, the launch of possible studies on recognition techniques applicable under the real time constraint has been set apart, at least for the moment.

6.3 Task 4: Sound Synthesis

This task was interrupted at a very preliminary stage. The only reliable conclusion that was drawn, was that under relatively low computational load the smartphone *could* play the role of a low-latency interface inside a closed-loop walking interaction paradigm.

Some oversimplified real-time sound synthesis models were enabled in the system in parallel with

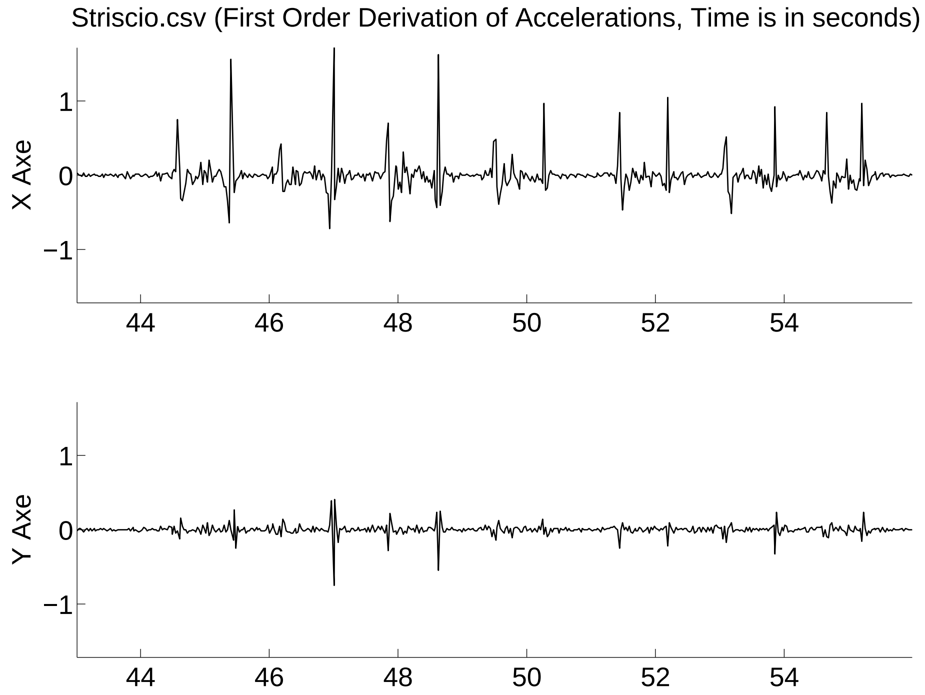


Figure 6.3: Example of data from the accelerometer, analyzed with Matlab (friction/drift).

the recording program. Among the sounds that were tested, the default phone *beep*, a discrete-time sinusoid whose frequency could be varied along time (video on `http://www.niwproject.eu/`), and finally white noise.

More precisely, sounds were produced following two strategies: i) by simply detecting acceleration thresholds, above which sound synthesis is enabled. This simple idea worked decently well to sonify scraping using white noise (video on `http://www.niwproject.eu/`), and ii) by mapping accelerations into sound synthesis parameters, either directly or by passing by a digital filter. In such sonification experiences, good results were obtained by first smoothing the data coming out from accelerometer with a lowpass filter, and then playing pure tones whose amplitude and frequency was driven by the x and y acceleration, respectively.

In spite of their incongruence with any real auditory cue of walking, such sounds turned useful to understand how to use the directionally dependent accelerometer information. Moreover, they proved that the delays and latencies of the entire analysis/synthesis system in use during this task were perceptually negligible.

Tasks 5 and 6 have been left pending. Currently they are inactive. They will be activated again if human resources will be found that can be allocated to this activity.

6.4 Conclusions

Clearly, the activity performed so far is far from providing satisfactory results and fulfilling the initial expected goals. In spite of this lack of completeness, some insight has been gained by the consortium about the possibility to design a simple interface for the augmentation of walking activities based on a cheap device such as the mobile phone.

Awareness has been gained about the fact that the more complex task in this activity consists in using the data from the accelerometer. The quality of the synthesis will ultimately depend on the quality of the analysis, and as far as we know no literature exists on the mapping of acceleration data into foot gestures. The person/month we have dedicated to uncover (at least partially) such maps using simple pattern recognition tools has not been sufficient to come to any robust conclusion.

At an informal level, we got the clear impression that the sole use of a cheap 3D accelerometer makes this analysis difficult and prone to error, especially in the detection of foot scraping. It would be not too difficult to pair this device with external force sensors. In fact, the audio DSP is full duplex in a way that a simple circuit can be realized to connect the sensors through the external microphone jack, without putting the hands on the hardware of the mobile phone.

In addition to that, we have experienced also some current unreliabilities of the software tools supporting the development of new applications for the Nokia N95. This tools in fact reveal several instabilities as soon as they are used for developing new, unconventional applications.

Ultimately, the results were encouraging. The simulated walk sensation is unrealistic, but not annoying as one could expect, due to the absence of perceived latencies in the response of the system. As a partial rehabilitation of the work done so far, the reader should be aware that many medical applications (both at state-of-the-art and research level) for rehabilitation, involving the use of auditory feedback in response to human movement, still make use of trivial sounds like beeps and pure tones [27]. It is an aim of this project to improve upon such a level.

7 Mobile Display of Walking Sounds

While thinking about ways to interactively display sounds to a user walking on an interface, capable of augmenting floors in the sense of the NIW project, at least three fundamental paradigms should come up to the mind of the sonic interaction designer:

- when the interaction occurs within a situated interface, for instance enabled by an active tile, sound is displayed by loudspeakers that can be incorporated into the floor;
- when the interaction is enabled instead by a mobile interface, say an augmented shoe, sound can be displayed by portable reproduction systems: either a pair of headphones or a set of body-mounted speakers.

These alternatives are dictated by the unreliability of a wireless communication, putting together pieces of a situated and a mobile interface when instantaneous feedback is required in a man-machine perception and action loop.

Nevertheless, the mentioned paradigms have profound differences, and making a scientifically consistent comparison between their acoustic features is far from being straightforward. Even if much is known about loudspeaker and headphone listening at both theoretical and practical implementation level [8], yet many acoustic facts related with such paradigms have not been systematically uncovered concerning the specific case of interactive walking. Among the many questions that an acoustician may address, some very reasonable ones are: i) how are sound sources perceived when they are reproduced by loudspeakers diffusing at ground level? ii) what is the quality of reproduced floor sounds through headphones? iii) is it better to enable high quality binaural listening of walking sounds using an expensive loudspeaker set capable of reproducing moving sound sources by means of some sophisticate diffusion model or, conversely, is it convenient to attach cheap speakers to the feet of a user?

While constantly keeping these issues in mind, also from an ergonomic viewpoint, the NIW project has not been designed to respond to the above questions. In fact, they would ask to conduct specific acoustic measurements followed by experiments on the auditory perception of floor sources, in parallel with the design and realization of sound reproduction sets that are tailored to use in floor- or shoe-based settings. NIW is not involved in loudspeaker or headphone research, design, and evaluation.

Rather, the project wants to help understand the realism and accuracy of a continuous foot-floor interaction once it is mediated by a machine displaying auditory cues of ground. This understanding moves through the experimental evaluation of user experiences, that will be delivered later and elsewhere during the project. In this deliverable section we report on the reproduction paradigms we have chosen, along with related technology.

7.1 Shoe- and body-mounted display

While prototyping pairs of augmented shoes (refer to Deliverable 3.1), UNIVR has looked for ways to obtain acceptable sound reproduction through small and lightweight, cheap loudspeaker-based technologies which did not heavily cluttered the interface hence preventing from a natural walking activity.

Figure 7.1 shows the first shoe mock-up made using rubber clogs. It makes use of a pair of 4Ω Visaton FRWS5 speakers back-ended with a handmade small acoustic chamber, made with paper and adhesive film in an aim to increase their frequency response toward the low frequency. The battery amplifier of a set of portable pc audio set was customized to provide the needed power to the speakers.

The following advantages [A] and limits [L] of this solution emerged:

[L] poor low frequency content along with disputable sound quality. All users informally reported lack of “smoothness” in the sounds. This problem is unavoidable as far as small size speakers are

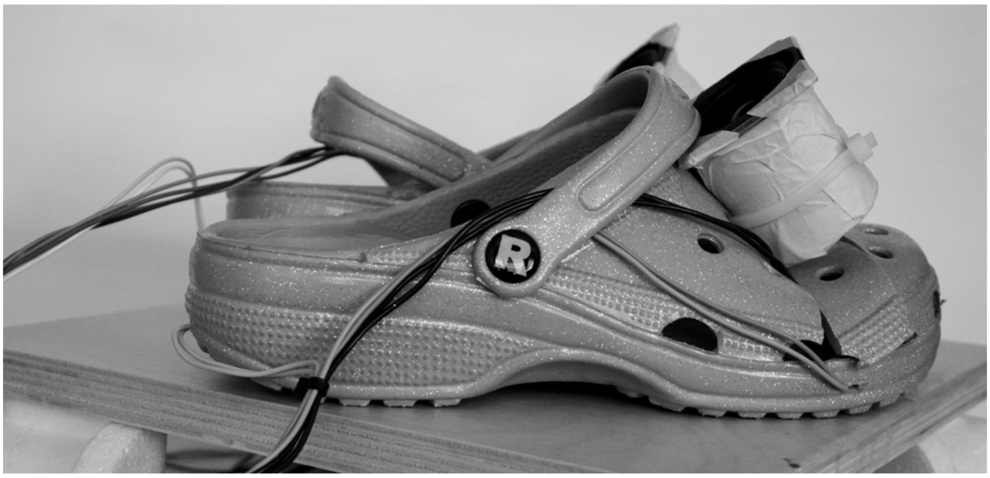


Figure 7.1: First shoe mock-up made with rubber clogs: speakers setting.

employed;

[A] accurate localization. This result is not so obvious: for instance, an early attempt to mount the loudspeakers in the rear part of the shoe resulted in sound displays "behind the legs" that did not provide satisfactory localization cues;

[L] low reproduction power. This drawback is caused by the limited power of most light battery amplifiers. It becomes of importance when the floor sounds are displayed in loud noise environments;

[L] overall "ecological" impression. Although not yet validated by rigorous experimental evaluations, all users found the shoe-based display natural and effective in spite of the fair sound quality.

The above limits were made less cogent in an improved setup, including an external ESI UGM-96 audio card in place of the resident sound device of the laptop, together with a pair of augmented sneakers (see Figure 7.2) in which the Visaton set was substituted with Goobay's Soundballs speakers, providing sound at lower frequency range and an internal amplifier that can be recharged offline.

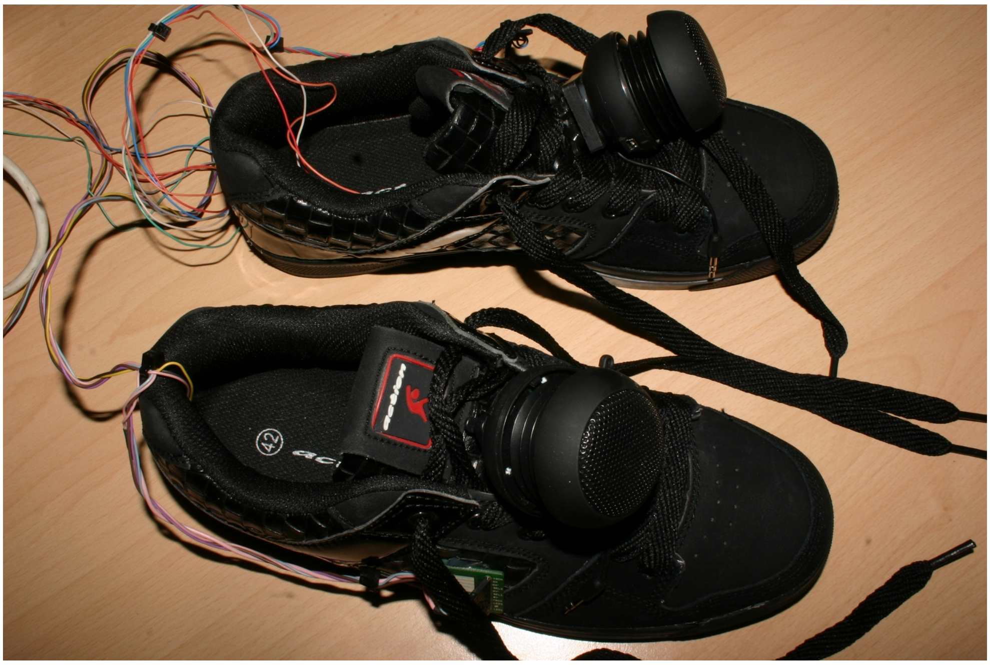


Figure 7.2: Second shoe mock-up made using sneakers: speakers setting.

In conclusion, the use of shoe-based speakers represents an interesting and perhaps necessary, but probably not sufficient step ahead in the research on interactive sonification of walking sounds.

For this reason, the consortium looked for ways to improve the existing reproduction setup.

7.2 Back-packed loudspeaker

The image of a walking person bringing a woofer in his or her back, that emits footstep sounds which are synchronized with the pace of this person, challenges the social conventions and, in some way, our common sense. In spite of the unusual appearance, a loudspeaker capable of adding low frequencies to the shoe-mounted small speaker set would improve the display a lot.

Objections may arise concerning the resulting quality of the auditory localization cues once the loudspeaker is included in the wearable set. By emitting sound at a different (i.e., not the floor) level, the new speaker unit in fact could have the effect to destroy the directionality of sound created by the shoe-mounted set. Fortunately, this effect can be avoided.

Low frequency sounds, in fact, determine lateralization cues in the form of interaural phase differences [6]. If the sound source is located in the vertical plane (that is, the plane that vertically cuts the human listener in two identical, symmetric parts) then these differences are not present in the perceived sound. In parallel, the elevation cues that are provided by the low frequencies are unreliable: they are easily overwhelmed by the more robust cues of elevation provided by the transient sounds coming from the shoe-mounted speakers. In conclusion, we can generate low frequencies that do not interfere with the spatial properties of a floor-based display meanwhile substantially improving the quality of the overall sound if this display is inherently poor in terms of low frequency content.

Thanks to existing technologies, we did not have to force a user to wear a large loudspeaker capable of producing low frequencies. Nowadays mini broadband loudspeakers in fact can be easily put within a backpack, furthermore they can be driven directly using the low-power output of an audio card thanks to buffers that retain energy in the system when the sound signal is not loud. In particular we have chosen the Yamaha NX-U02 loudspeaker unit (see Figure 7.3), that picks up the audio and needed power from a USB connection to the pc, and can reproduce signals up to about 30 W.

By means of this device we have achieved sufficient loudness and spectral content of the walking sounds, without affecting localization elicited by the perceived auditory cues of floor. By passing the output from the synthesis models through a digital cross-over, capable of splitting the sound into low and high frequency bands, separate signals have been respectively delivered to the portable unit and the shoe-based speakers.

Current experiences using this setup suggest that this mobile loudspeaker set is capable of providing accurate auditory display without limiting a user's normal walking activity. We will use this set for performing experiments during the next project period.

7.3 Auditory stimuli through underfoot tactile feedback

It is known that loudspeakers generate vibrations that can be perceived by touching their cabinet, or at least the transducer—the implications of this effect to headphone listening are indeed of major importance [8]. Since making use of the same voice coil-based technology although differently arranged, not surprisingly the vibrotactile transducers the project has employed for the tiles (refer also to Deliverable 2.1, 3.1, and Chapter 8) generate sound as a side effect.

Provided that the perceived frequencies in touch are far lower compared to hearing, the active band of such transducers decays after few kHz, and nevertheless it results in clearly audible and meaningful sounds in that band once the "loudness" is turned up. For this reason, vibrotactile feedback actuated by voice coils underfoot represents a powerful technique for the enabling of a strongly coherent, directionally robust acusto-haptic display. This technique is expected to turn useful for the forthcoming experiments, especially those involving everyday sounds whose high frequency content does not influence their ecological meaning.

Figure 7.3: Portable loudspeaker.

Conversely, the small speakers that are mounted on the shoes do not produce any tactile feeling, as their frequency content is largely off the perceptual band for the touch modality, and their loudness is comparably low. Nor the back-packed loudspeaker is sufficiently coupled with the skin to convey similar sensations to the body. For this reason, additional haptic feedback underfoot will be needed to perform multimodal experiments involving the mobile foot-based interface setting.

8 Multimodal Synthesis of Material Fracture Underfoot

This chapter of the deliverable describes research in the Shared Reality Environments Lab at McGill University on the multimodal synthesis of brittle fracture underfoot in an immersive virtual environment context. The methods developed in this research were designed to enable interactive synthesis of auditory and vibrotactile signals, tailored to the setting of audio and vibrotactile display via a rigid floor component, of the type described in relevant sections of Deliverables 3.1 and 2.1.

A hybrid approach to the rendering and display problem was adopted, consisting of a lumped (time-domain only) model of a stochastic elasto-plastic process governing the local, spatially-averaged properties of the material response, together with a distributed model for the spatial propagation of fracture lines in the material, driven in time by the stochastic temporal process. Interaction capture was accomplished via a distributed floor surface, using the contact-based force sensing methods described in Deliverable 3.1.

As an engaging and interactive virtual environment scenario applying these techniques, we implemented a demonstration allowing its users to walk over a virtual pond whose surface has frozen over. If a user walks on the pond insufficiently carefully, patterns of cracks form in the surface ice, dependent on the locations and forces of foot-floor contact. The interactions with this virtual material are rendered using audio, visual, and vibrotactile channels.

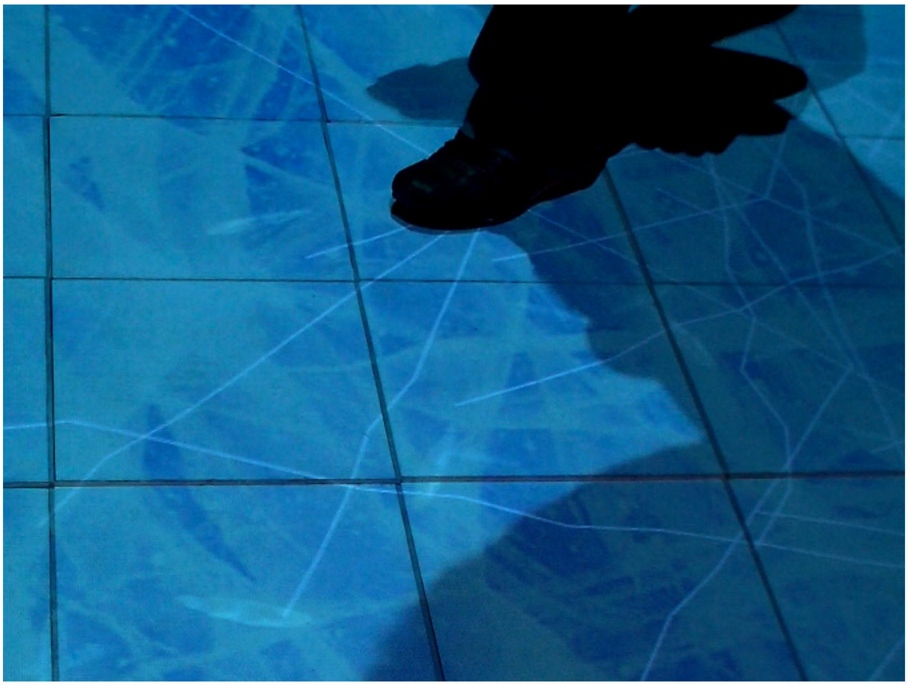


Figure 8.1: Still image from video documentation of an illustrative interactive scenario in which the multimodal synthesis and contact sensing techniques described in this deliverable, and in D3.1, are used to allow users to walk over a virtual pond whose surface has frozen over. Cracks in the virtual ice sheet dynamically form at loci of contact between feet and floor.

The results are presented in detail in a report that is included in the supplementary materials for this deliverable in `www.niwproject.eu`, in the area related to this chapter. (The report itself is currently undergoing blind peer review for a conference proceedings, so should not be treated as public material in its present form.) Audiovisual documentation of the application of these techniques to ice fracturing underfoot can be retrieved from the same area.

References

[1] S. aus der Wiesche. Computational slosh dynamics: theory and industrial application. *Computational Mechanics*, 30:374–387, 2003.

[2] F. Avanzini and P. Crosato. Integrating physically based sound models in a multimodal rendering architecture: Research articles. *Comput. Animat. Virtual Worlds*, 17(3-4):411–419, 2006.

[3] F. Avanzini and D. Rocchesso. Modeling collision sounds: non-linear contact force. In *Proc. Conf. on Digital Audio Effects (DAFX-01)*, pages 61–66, Limerick, December 2001.

[4] F. Avanzini and D. Rocchesso. Physical modeling of impacts: theory and experiments on contact time and spectral centroid. In *Proc. Int. Conf. on Sound and Music Computing (SMC'04)*, pages 287–293, 2004.

[5] S. Bilbao. Robust physical modeling sound synthesis for nonlinear systems. *Signal Processing Magazine, IEEE*, 24(2):32–41, March 2007.

[6] J. Blauert. *Spatial Hearing: the Psychophysics of Human Sound Localization.* MIT Press, Cambridge, MA, 1983.

[7] G. Borin, G. De Poli, and D. Rocchesso. Elimination of Delay-Free Loops in Discrete-Time Models of Nonlinear Acoustic Systems. *IEEE Trans. on Speech and Audio Processing*, 8(5):597–605, September 2000.

[8] J. Borwick, editor. *Loudspeaker and Headphone Handbook.* Focal Press, Oxford (UK), 1998.

[9] R. Bresin, S. D. Monache, F. Fontana, S. Papetti, P. Polotti, and Y. Visell. Auditory feedback from continuous control of crumpling sound synthesis. In *CHI 2008 Workshop on Sonic Interaction Design*, Florence, Italy, Apr. 5-10 2008. ACM.

[10] A. Chaigne and J. Kergomard. *Acoustique des Instruments de Musique.* Belin, 2008.

[11] J. E. Colgate and J. M. Brown. Factors Affecting the Z-Width of a Haptic Display. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 3205–3210, San Diego, May 1994.

[12] P. R. Cook. Physically informed sonic modeling (PHISM): Synthesis of percussive sounds. *Computer Music Journal*, 21(3):38–49, 1997.

[13] P. R. Cook. Modeling Bill's gait: Analysis and parametric synthesis of walking sounds. In *Proc. Audio Engineering Society 22 Conference on Virtual, Synthetic and Entertainment Audio*, Espoo, Finland, July 2002. AES.

[14] T. Delbruck, A. M. Whatley, R. Douglas, K. Eng, K. Hepp, and P. F. M. J. V. Verschure. A tactile luminous floor for an interactive autonomous space. *Robotics and Autonomous Systems*, 55:433–443, 2007.

[15] N. Diolaiti, C. Melchiorri, and S. Stramigioli. Contact impedance estimation for robotic systems. *IEEE Transactions on Robotics*, 21(5):925–935, 2005.

[16] P. Dupont, V. Hayward, B. Armstrong, and F. Altpeter. Single state elastoplastic friction models. *IEEE Transactions on Automatic Control*, 47(5):787–792, 2002.

[17] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 736–744, New York, NY, USA, 2002. ACM.

[18] A. J. Farnell. Marching onwards – procedural synthetic footsteps for video games and animation. In *pd Convention*, 2007.

[19] P. Flores, J. Ambrósio, J. Claro, and H. Lankarani. Influence of the contact-impact force model on the dynamic response of multi-body systems. *Proceedings of the Institution of Mechanical Engineers-K*, 220(1):21–34, 2006.

[20] P. Flores, J. P. Claro, and H. M. Lankarani. *Kinematics and Dynamics of Multibody Systems with Imperfect Joints: Models and Case Studies.* Springer, 2008.

[21] G. J. Franz. Splashes as sources of sound in liquids. *The Journal of the Acoustical Society of America*, 31(8):1080–1096, 1959.

[22] W. W. Gaver. How Do We Hear in the World? Explorations in Ecological Acoustics. *Ecological Psychology*, 5(4):285–313, Apr. 1993.

[23] V. Hayward. Physically-based haptic synthesis. In *Haptic Rendering: Foundations, Algorithms and Applications.*

[24] K. H. Hunt and F. R. E. Crossley. Coefficient of restitution interpreted as damping in vibroimpact. *ASME J. Applied Mech.*, pages 440–445, June 1975.

[25] G. Kuwabara and K. Kono. Restitution coefficient in a collision between two spheres. *Jap. J. of Appl. Phys.*, 26(8):1230–1233, 1987.

[26] H. M. Lankarani and P. E. Nikravesh. A contact force model with hysteresis damping for impact analysis of multibody systems. *Journal of Mechanical Design*, 112(3):369–376, 1990.

[27] M.-Y. Lee, K.-S. Soon, and C.-F. Lin. New computer protocol with subsensory stimulation and visual/auditory biofeedback for balance assessment in amputees. *J. of Computers*, 4(10):1005–1011, Oct. 2009.

[28] D. W. Marhefka and D. E. Orin. A compliant contact model with nonlinear damping for simulation of robotic systems. *IEEE trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 29(6):566–572, November 1999.

[29] S. L. Means and R. M. Heitmeyer. Low-frequency sound generation by an individual open-ocean breaking wave. *The Journal of the Acoustical Society of America*, 110(2):761–768, 2001.

[30] M. Minnaert. On musical air bubbles and the sounds of running water. *Philos. Mag.*, 16:235–248, 1933.

[31] S. D. Monache, D. Devallez, C. Drioli, F. Fontana, S. Papetti, P. Polotti, and D. Rocchesso. Sound synthesis tools for sound design. Deliverable 2.2, CLOSED project, 2008. Available at http://closed.ircam.fr.

[32] M. Muller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. *In Proc. ACM SIGGRAPH/Eurographics 03*, pages 154–159, 2003.

[33] M. Müller, J. Stam, D. James, and N. Thürey. Real time physics: class notes. In *SIGGRAPH '08: ACM SIGGRAPH 2008 classes*, pages 1–90, New York, NY, USA, 2008. ACM.

[34] J. A. Nystuen, J. Leo H. Ostwald, and H. Medwin. The hydroacoustics of a raindrop impact. *The Journal of the Acoustical Society of America*, 92(2):1017–1021, 1992.

[35] S. O'Modhrain and G. Essl. PebbleBox and CrumbleBag: Tactile interfaces for granular synthesis. In *Proc. NIME2004.*

[36] S. Papetti, F. Avanzini, and D. Rocchesso. Energy and accuracy issues in numerical simulations of a non-linear impact model. In *Proc. Conf. on Digital Audio Effects (DAFX-09)*, Como, Italy, September 2009.

[37] L. Peltola, C. Erkut, P. R. Cook, and V. Välimäki. Synthesis of hand clapping sounds. *IEEE Trans. Audio, Speech and Language Proc.*, 15(3):1021–1029, 2007.

[38] M. Puckette. Pure Data. In *Second Intercollege Computer Music Concerts*, pages 37–41, Tachikawa, Japan, 1996. Available at `http://www-crca.ucsd.edu/~msp/`.

[39] H. C. Pumphrey, L. A. Crum, and L. Bjorno. Underwater sound produced by individual drop impacts and rainfall. *The Journal of the Acoustical Society of America*, 85(4):1518–1526, 1989.

[40] L. Pust and F. Peterka. Impact oscillator with Hertz's model of contact. *Meccanica*, 38(1):99–116, 2003.

[41] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics.* Springer, 2nd edition, 2007.

[42] M. Rath and D. Rocchesso. Continuous sonic feedback from a rolling ball. *IEEE Multimedia*, 12(2):60–69, 2005.

[43] M. Rein. Phenomena of liquid drop impact on solid and liquid surfaces. *Fluid Dynamics Research*, 12(2):61–93, 1993.

[44] D. Rocchesso. *Introduction to Sound Processing.* Mondo Estremo, Florence, Italy, 2003. Online available at http://www.faqs.org/docs/sp/.

[45] D. Rocchesso, R. Bresin, and M. Fernström. Sounding objects. *IEEE MultiMedia*, 10(2):42–52, 2003.

[46] D. Rocchesso and F. Fontana, editors. *The Sounding Object.* Mondo Estremo, 2003. Available at http://www.soundobject.org/.

[47] K. van den Doel. Physically-based models for liquid sounds. *ACM Transactions on Applied Perception*, 2(4):534–546, 2005.

[48] K. van den Doel, P. Kry, and K. Pai. Foleyautomatic: Physically-based sound effects for interactive simulation and animation. *In Proc. ACM SIGGRAPH 01*, pages 537–544, 2001.

[49] L. Verlet. Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Physical Review*, 159(1):98+, July 1967.

[50] Y. Visell, J. Cooperstock, B. L. Giordano, K. Franinovic, A. Law, S. McAdams, K. Jathal, and F. Fontana. A vibrotactile device for display of virtual ground materials in walking. In *Proc. of Eurohaptics 2008*, 2008.

[51] Y.Dobashi, T.Yamamoto, and T.Nishita. Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics. *ACM Trans. on Graphics*, 22(3 (Proc. SIGGRAPH2003)):732–740, 2003.