

APPENDIX A COMPUTING THE HIERARCHICAL WEIGHTS

Algorithm 1 describes a simple way to compute the hierarchical weights β_{ij} from the skinning weights α_{ij} .

Algorithm 1 Computing hierarchical weights β_{ij}

```

for  $i \in \mathcal{I}$  do
  Initialize  $\forall j \in \mathcal{J}, \beta_{ij} = 0$ 
  for  $j_0$  such that  $\alpha_{ij_0} > 0$  do
     $\beta_{ij_0} = \alpha_{ij_0}$ 
     $j = j_0$ 
    while  $j > 0$  do
       $\beta_{ia(j)} = \beta_{ij} + \alpha_{ia(j)}$ 
       $j = a(j)$ 
    end while
  end for
end for

```

APPENDIX B INTEGRATION BY PART FROM EQ. 14

We show that

$$\int_{u=0}^W \dot{\kappa}(u) \Omega_j(t-u) du = \int_{u=0}^W \kappa(u) \dot{\Omega}_j(t-u) du$$

Integrating by part the left hand side leads to

$$\int_{u=0}^W \dot{\kappa}(u) \Omega_j(t-u) du = \underbrace{[\kappa(u) \Omega_j(t-u)]_0^W}_{\simeq 0} - \int_0^W \kappa(u) (-\dot{\Omega}_j(t-u)) du,$$

The first term is negligible as we assume $\kappa(0) = 0$, and $\kappa(W) \simeq 0$, which leads to the desired result.

APPENDIX C KERNEL ALTERNATIVES

We describe in the following three examples of possible custom procedural oscillation kernels κ_1 , κ_2 , and κ_3 illustrated in Fig. 6.

First, we can represent a time-varying frequency in considering

$$\kappa_1(t) = \sin(2\pi(f_0 + f_s t)t),$$

where f_0 represents the initial frequency and f_s is the slope of a linear frequency increase. This kernel models a vibration featuring low frequency at the start of the motion (i.e. large magnitude), and higher frequency in its end (i.e. low magnitude). The effect of this mimics an object's response to disturbance which damps out but appears to do so with faster and smaller oscillations.

Second, we consider a kernel representing oscillation with long-lasting maxima mimicking the type of vibration observed on pendulum with large angle. We formulated this effects in considering the first harmonics of a trapezoid as

$$\kappa_2(t) = \frac{1}{1+1/9+1/25} (\sin(2\pi ft) + \sin(6\pi ft)/9 - \sin(10\pi ft)/25).$$

Third, we consider the extreme case of a triangular response which convey a sense of a more artificial mechanical like oscillation. In such a case, we set

$$\kappa_3(t) = \text{triangle}(f, t),$$

where $\text{triangle}(f, t)$ is the triangle function of frequency f between $[-1, 1]$.

APPENDIX D USER-DEFINED PARAMETERS

We describe in this appendix the parameters and the default values that can be used to tune the results.

Table 1: Input data associated with the model

Parameter	Role	Value
α_{ij}	Skinning weights	User defined (rig)
β_{ij}	Hierarchical weights	Eq. (6)

D.1 Input data

We first present in table 1 the input data attached to the rigged model that represents the spatial extent of the deformation associated with the joints.

D.2 DS default parameters

Table 2 presents the default parameters of the DS model. All these parameters can be modified straightforwardly by the user in order to adapt the general deformation and oscillation effects. We assume a model to be of unit size (1 m) for size-related parameter values.

Table 2: Standard DS parameters, their roles, and typical values.

Parameter	Role	Typical value
Bend effect ψ^{bend} , see Sec. 3.2.1		
γ^{bend}	Bending magnitude	1
Oscillation kernel κ , see Sec. 4		
A	Oscillation magnitude	0.3
τ	Oscillation damping	1 (s)
f	Oscillation frequency	1.5 (Hz)
W	Convolution time windows	5τ (s)
Time delay , see Sec. 5.1		
c	Wave propagation speed	1 (m/s)

D.3 Advanced fine-tuning

Finally, the DS deformation can be fine-tuned along the following possibilities

- Adding a per-vertex and/or per-joint dependence to the parameters $\gamma_{ij}^{\text{bend}}, A_{ij}, \tau_{ij}, f_{ij}$, see Sec. 6.1.
 - A per-vertex dependence, that can be manually painted, allows to fine-tune the deformation beyond the level of precision of the rig.
 - A per-joint dependence allows for treating each joint-related deformation separately and relying on the smoothness and expressivity of the skinning weights.
- Setting a varying deformation origin via the Flexible Origine Operator $O_{j_{\min}}$, allowing to represent the effect of a flexible limb attached to a more rigid body part, see Sec. 6.3.

APPENDIX E PSEUDO CODE FOR DS COMPUTATION

We describe in Algorithm 2 corresponding to the DS skinning deformation, including the use of the procedural oscillation effect and bend deformer.

Algorithm 2 Computing Dynamic Skinning deformation

```

1: # Oscillation effect
2: function  $\kappa(t)$ 
3:   return  $A \exp(-t/\tau) \sin(2\pi ft)$   $\triangleright$  Eq. (11)
4:   # Or any alternative kernel function as in Appendix C.
5: end function

6: # Bend deformer
7: function  $\psi^{\text{bend}}(\Omega, u)$ 
8:   axis =  $\Omega / \|\Omega\|$ , angle =  $-\gamma^{\text{bend}} \|\Omega \times u\|$ 
9:   build  $R = \text{Rotation}(\text{axis}, \text{angle})$ 
10:  return  $R u$   $\triangleright$  Eq.(9)
11: end function

12:  $\forall i \in \mathcal{I}, \forall j \in \mathcal{J}$ , Initialize  $\beta_{ij}$   $\triangleright$  Appendix A
13: Initialize  $\forall j \in \mathcal{J}, L_{\Omega_j} = []$  # Stored angular velocity
14: # Main animation loop
15: for  $m = 0$ ; while true do
16:   Initialize  $\forall j \in \mathcal{J}, L_{\Omega_j[m]} = \Omega_j$ 
17:    $\forall i \in \mathcal{I}, p_i^{\text{LBS}} = \text{LBS}(p_i)$  # Linear Blend Skinning
18:   for all  $i \in \mathcal{I}$  do
19:     Initialize  $\text{DS}_{\text{deform}} = 0$ 
20:     for all  $j \in \mathcal{J}$  do
21:       # Numerical convolution
22:        $\Omega_{\text{filtered}} = L_{\Omega_j[m]}$ 
23:       for  $n = 0$ ; while  $n < W/\Delta t$  do
24:          $\Delta\Omega_j = L_{\Omega_j[m-n]} - L_{\Omega_j[m-n-1]}$ 
25:          $\Omega_{\text{filtered}} += \kappa(n\Delta t) \Delta\Omega_j$   $\triangleright$  Eq.(15)
26:          $n = n + 1$ 
27:       end for
28:       # DS deformation for joint j at vertex i
29:        $\text{DS}_{\text{deform}} += \beta_{ij} \psi^{\text{bend}}(\Omega_{\text{filtered}}, p_i^{\text{LBS}} - q_j)$ 
30:     end for
31:      $p_i^{\text{LBS}} = p_i^{\text{LBS}} + \text{DS}_{\text{deform}}$   $\triangleright$  Eq.(8)
32:   end for
33:    $m = m + 1$ 
34: end for

```
