

Audio cross-fade implementation via rational functions

Lucian Lupşa-Tătaru

Faculty of Electrical Engineering and Computer Science,
“Transilvania” University of Braşov, Bd. Eroilor No. 29, Braşov, Romania,
lupsa@programmer.net, lucian.lupsa@unitbv.ro

Abstract. Implementation of the cross-fade audio effect requires shaping the fade profile for a certain audio content that is to be faded out, as well as customizing the audio fade for an additional audio content, which is to be faded in, with the purpose of achieving a smooth transition between the two different audio contents. Similar to the case of applying adjustable fades, the audio cross-fades are usually carried out in the off-line mode, by employing various transcendental functions to shape the audio fades (both fade-out and fade-in effects). To improve the computational capabilities by minimizing the delay between receiving the position within the cross-fade effect and returning both the volume of the faded out section and the volume of the faded in one, we consider that, during the cross-fade effect, the audio volume of each of the two overlapping sections is the output of a rational function i.e. a mapping defined by a rational fraction. A plain HTML5/JavaScript implementation, prepared to be tested in any major browser, is advanced in the paper in order to highlight the suitability of the suggested approach to audio cross-fade customization with real-time computing.

Keywords. Real-time computing, audio fade-out, audio fade-in, audio cross-fade, HTML5, JavaScript, programming techniques.

1. Introduction

In audio engineering, the cross-fade effect is related to audio volume processing. It is well-known and widely used with the purpose of carrying out a smooth transition between two separate audio contents. Performing a cross-fade engages the simultaneous application of fade-out and fade-in audio effects so that the transition from the first content to the second one sounds more natural than is without inserting silence. More precisely, during the same time interval, the first audio content is faded out while the second audio content is faded in. Since the length of the fade-out effect to be applied upon the first audio content is identical to the length of the fade-in effect to be applied upon the second audio content, one identifies a faded out audio section that overlaps a faded in section [1]-[6].

Practically, customizing the audio cross-fade involves setting out the length of the overlapping region, which comes to be the cross-fade length, as well as shaping the fades to be applied upon the two overlapping audio sections [3]-[6]. Since the audio fades are traditionally implemented by means of transcendental functions (e.g. logarithm, exponential, sine) to impose the time-related evolution of the audio volume, the implementation of the audio cross-fade is mostly performed within audio editors by constructing and concurrently valuating two distinct transcendental functions.

However, with the increasing emphasis placed on reactive computing [7]-[15], the primary concern is now related to optimizing the delay between receiving the input, i.e. the playback position, and returning the output, i.e. the audio volume of a definite faded section. In this context, it has been

proven that the employment of particular rational functions to shape the audio fades provides both effectiveness and high versatility [16], [17]. Hence, in order to maintain the suitability with reactive computing, the cross-fade implementation will be carried out by considering that the volume of each of the two overlapping sections represents precisely the output of an appropriate rational function.

2. Cross-fade customization

To fade the first audio content out, we consider the following rational function that associates the current position within the audio cross-fade to the audio volume [16]:

$$v_o(\tau) = \frac{\tau^k - \alpha}{\beta\tau^k - \gamma}, \tau \in [0, \tau_f], \quad (1)$$

$$k \in \{1, 2, 3, 4\}$$

wherein one identifies [16], [17]:

v_o - the mapping output i.e. the volume of the first audio content, which is to be faded out,
 τ - the position within the cross-fade effect i.e. the difference of the playback position within the first audio content and the position corresponding to cross-fade initiation (the mapping input),
 τ_f - the audio cross-fade length.

Both the high level of versatility and the suitability with reactive computing, received by implementing rational function (1) to customize the fade-out transition, have been emphasized in detail [16], [17]. It can also be observed that, by adopting (1), the resulted fade-out effects can act, in real-time, similar to the fade-outs of exponential, logarithmic and S-curve shape, respectively [16].

Having at hand function (1), wherein alongside parameter k , the coefficients α , β and γ decide the fade-out profile, in order to preserve the suitability with real-time computing, we adopt the following composite function with a view to fading the second audio content in:

$$v_I(\tau) = (v_o \circ \delta)(\tau), \tau \in [0, \tau_f], \quad (2)$$

$$\delta(\tau) = \tau_f - \tau$$

that is

$$v_I(\tau) = v_o(\tau_f - \tau) = \frac{(\tau_f - \tau)^k - \alpha}{\beta(\tau_f - \tau)^k - \gamma}, \tau \in [0, \tau_f], \quad (3)$$

$$k \in \{1, 2, 3, 4\}.$$

With rational function (1) employed to fade the first audio content out, and rational function (3) used to fade the second audio content in, one perceives that at the cross-fade midpoint $\tau_h = \tau_f/2$, the audio volumes come to be exactly the same, i.e.

$$v_o(\tau_h) = \frac{\tau_h^k - \alpha}{\beta\tau_h^k - \gamma} = \frac{(\tau_f - \tau_h)^k - \alpha}{\beta(\tau_f - \tau_h)^k - \gamma} = v_I(\tau_h). \quad (4)$$

Hence, one may introduce:

$$v_h = v_o(\tau_h) \equiv v_I(\tau_h). \quad (5)$$

For $\tau = 0$ in (3), we have

$$v_I(0) = v_O(\tau_f) = \frac{\tau_f^k - \alpha}{\beta\tau_f^k - \gamma}. \quad (6)$$

Relationship (6) plainly reveals that the volume of the second audio content (to be faded in) at the beginning of the cross-fade effect is identical to the volume of the first audio content (to be faded out) at the end of cross-fading. On the other hand, according to relation (3), for $\tau = \tau_f$ we get:

$$v_I(\tau_f) = v_O(0) = \alpha/\gamma. \quad (7)$$

Thus, the employment of the composite function (2) ensures that, at the end of cross-fading, the volume of the second audio content, which is to be faded in, matches the volume of the first audio content at the cross-fade initiation i.e. the maximum volume.

Imposing now the value v_m for the volume of the first audio content at the beginning of the cross-fade that is precisely the volume of the second audio content at the end of cross-fading, and taking into account that the fade-out effect leads to silence while the fade-in transition starts out from silence, one may set:

$$\begin{aligned} v_O(0) = v_m &\Leftrightarrow v_I(\tau_f) = v_m, \\ v_O(\tau_f) = 0 &\Leftrightarrow v_I(0) = 0. \end{aligned} \quad (8)$$

On the other hand, according to (5), we have the audio volumes at the cross-fade halfway point:

$$\begin{aligned} v_O(\tau_h) &\equiv v_I(\tau_h) = v_h = \rho_h v_m, \\ \rho_h &= v_h/v_m; \quad 0 < \rho_h < 1 \end{aligned} \quad (9)$$

wherein quantity ρ_h stands for the ratio of the volume of the first or of the second audio content, occurring at the cross-fade midpoint, to the initial volume of the first audio content, which is to be faded out, that is just the final volume of the second audio content, which is to be faded in.

Having in view relationship (1) along with conditions (8) and (9), one receives the coefficients interfering both in rational fraction (1) and rational fraction (3), which yield the audio contents levels during the cross-fade [16]:

$$\begin{aligned} \alpha &= \tau_f^k, \\ \beta &= \frac{1 - 2^k(1 - \rho_h)}{\rho_h v_m}, \\ \gamma &= \tau_f^k/v_m. \end{aligned} \quad (10)$$

Furthermore, relation (1), where the coefficients α , β , γ are given by (10), validates the decaying of the volume of the first audio content during the cross-fade effect. More precisely, (1) and (10) lead to a negative rate of change of the volume of the first audio content [16]:

$$\frac{dv_O}{d\tau} = -k \frac{\tau^{k-1}}{(\beta\tau^k - \gamma)^2} (\gamma - \alpha\beta), \quad k \in \{1, 2, 3, 4\} \quad (11)$$

wherein, taking into account (9),

$$\gamma - \alpha\beta = \frac{\tau_f^k}{v_m} (2^k - 1) \left(\frac{1}{\rho_h} - 1 \right) > 0. \quad (12)$$

Similarly, according to relation (2), defining the composite function that associates the position within the cross-fade effect to the volume of the second audio content, one obtains:

$$\frac{dv_I}{d\tau} = k \frac{\delta^{k-1}}{(\beta\delta^k - \gamma)^2} (\gamma - \alpha\beta) = k \frac{(\tau_f - \tau)^{k-1}}{[\beta(\tau_f - \tau)^k - \gamma]^2} (\gamma - \alpha\beta). \quad (13)$$

Taking now into consideration (12), it plainly follows that (2) accurately depicts a fade-in audio effect, i.e. we have

$$\frac{dv_I}{d\tau} > 0, \tau \in (0, \tau_f). \quad (14)$$

With mappings (1) and (3), employed for implementing audio fade-outs and fade-ins, respectively, customizing the cross-fade implies setting the value of the cross-fade length i.e. τ_f , selecting k out of the set $\{1, 2, 3, 4\}$, and imposing the value of quantity ρ_h i.e. the ratio of the volume of the faded out section, occurring at the halfway point, to the volume of the same audio section, detected at the cross-fade initiation. The high level of versatility of the cross-fade effect received by implementing (1) and (3) is due to the fact that, according to relations (11)-(14), we have

$$\frac{dv_O}{d\tau} < 0 \text{ and } \frac{dv_I}{d\tau} > 0, \tau \in (0, \tau_f)$$

for any k within the set $\{1, 2, 3, 4\}$, and for any value of ρ_h , defined by (9), situated in the open interval $(0, 1)$ that is for any volume of each of the two overlapped audio sections, which is imposed at the cross-fade midpoint and is less than the initial volume of the faded out section.

To emphasize the versatility of the cross-fade effect obtained by implementing rational functions (1) and (3) in order to perform fade-out and fade-in transitions, we have portrayed the curves of Fig. 1 up to Fig. 4. Each figure corresponds to a certain value of parameter k in relations (1), (3) and depicts a set of 5 s cross-fade effects, received for quantity ρ_h in the set $\{0.2, 0.5, 0.7\}$.

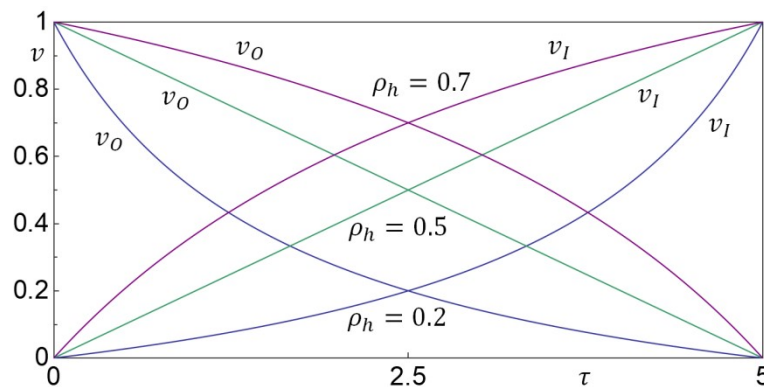


Fig. 1. Illustration of a set of three cross-fades, received for $k = 1$ in mappings (1), (3) and quantity $\rho_h \in \{0.2, 0.5, 0.7\}$.

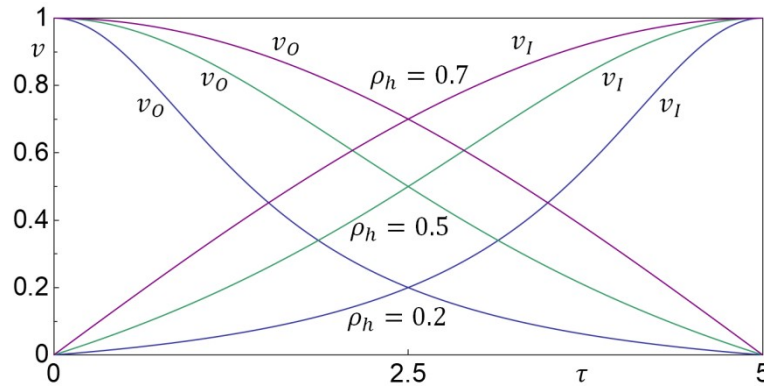


Fig. 2. Illustration of a set of three cross-fades, received for $k = 2$ in mappings (1), (3) and quantity $\rho_h \in \{0.2, 0.5, 0.7\}$.

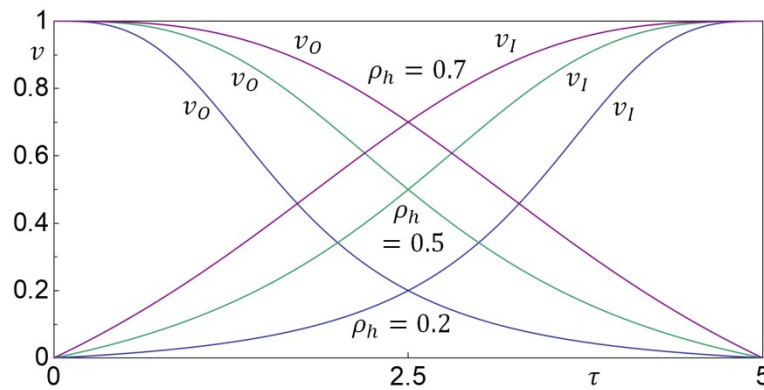


Fig. 3. Illustration of a set of three cross-fades, received for $k = 3$ in mappings (1), (3) and quantity $\rho_h \in \{0.2, 0.5, 0.7\}$.

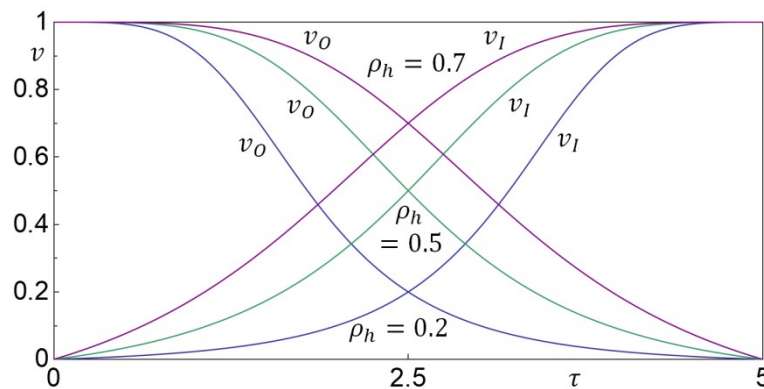


Fig. 4. Illustration of a set of three cross-fades, received for $k = 4$ in mappings (1), (3) and quantity $\rho_h \in \{0.2, 0.5, 0.7\}$.

As forenamed, according to (9), quantity ρ_h denotes the ratio of the volume of the audio sections, which is imposed at the cross-fade midpoint, to the maximum volume of the audio sections. For the sake of simplicity, we have assumed that the maximum volume interfering in Fig. 1 up to Fig. 4 has the value $v_m = 1$. In this case, one perceives that quantity ρ_h , introduced by (9), comes to be just the audio volume v_h , imposed at the cross-fades halfway point that is $\tau_h = 2.5$ s.

One observes that the cross-fade effect of Fig. 1, which corresponds to $k = 1$ and $\rho_h \equiv v_h = 0.5$, is carried out by applying linear fades. On the other hand, the cross-fades corresponding to $\rho_h = 0.5$ and represented in Fig. 2, Fig. 3, and Fig. 4, received by successively increasing the value of k in mappings (1) and (3), are performed via fades that incorporate attributes of the S-curve shape [4]-[6]. Thus, even if ρ_h , i.e. the volume at the midpoint, is preserved at the value of 0.5, in contrast to the effect depicted in Fig. 1, the cross-fades of Fig. 2, Fig. 3 and Fig. 4 are carried out by maintaining the faded out sections at a higher level in the beginning of the effect and at a lower level towards its ending end, at the same time, by keeping the faded in sections at a lower level in the beginning of the cross-fade and at a higher level in the ending region of the effect. The same applies to the cross-fades corresponding to $\rho_h = 0.2$ and $\rho_h = 0.7$, respectively, having in view that not only in the beginning but also in the ending region of a fade-out received by implementing (1), the absolute value of the rate of change of the audio volume decreases with parameter k , encompassed by fraction (1) [16] whilst both in the beginning and in the ending region of a fade-in received by implementing the composite function (3), the rate of change of audio volume decreases with k in (3).

3. JavaScript implementation

In order to check the suggested technique of customizing audio cross-fades from the perspective of reactive computing, we have prepared an HTML5/JavaScript implementation. Assuming that the locators (URLs) of the audio files are provided by the user in the form of function arguments, the code advanced here is ready for running in any major browser. As it is designed, the application initiates a 5 s cross-fade at the playback position of 20 s within the audio contents. To prove the efficiency of the technique, we have set the parameters values, i.e. the functions arguments, so that the cross-fade corresponds to $\rho_h = 0.7$ in Fig. 4 that is for $k = 4$ in mappings (1) and (3). On the other hand, with the purpose of accomplishing discretization, the “setInterval()” method of the “window” object is employed in order to evaluate the outputs of (1) and (3), wherein $k = 4$, once every 50 ms.

The code of the described application is put forward in the following lines.

```
<!DOCTYPE html>
<html>
<head>
  <title>Cross-fade</title>
</head>
<body>
<script>

var audE1, audE2, butte; // audE1, audE2: audio objects, butte: button object
var alpha, beta, gamma; // hold the coefficients in mappings (1), (3)
var timerId; // id returned by the setInterval() method
// global scope
function compCoeff( tauF, rhoH, k ) {
  // computes the coefficients in (1) and (3) for max. volume = 1
  switch( k ) {
    case 1:
      alpha = tauF;
      beta = 2.0 - 1.0 / rhoH;
      break;
    case 2:
      alpha = tauF * tauF;
      beta = 4.0 - 3.0 / rhoH;
      break;
    case 3:
      alpha = tauF * tauF * tauF;
      beta = 8.0 - 7.0 / rhoH;
      break;
    default: // assumes parameter k = 4
      alpha = tauF * tauF; alpha = alpha * alpha;
      beta = 16.0 - 15.0 / rhoH;
  }
  gamma = alpha; // valid for max. volume = 1
}
```

```

function v( t, k ) {
  var tToThek;
  switch( k ) {
    case 1:
      tToThek = t;
      break;
    case 2:
      tToThek = t * t;
      break;
    case 3:
      tToThek = t * t * t;
      break;
    default:
      // assumes parameter k = 4
      tToThek = t * t; tToThek = tToThek * tToThek;
  }
  var vol = ( tToThek - alpha ) / ( beta * tToThek - gamma );
  return vol;
}

function v0( tau, k ) {
  // returns the output of mapping (1)
  var volOut = v( tau, k );
  return volOut;
}

function vI( tau, tauF, k ) {
  // returns the output of mapping (3)
  var delta = tauF - tau;
  var volIn = v( delta, k );
  return volIn;
}

function setVols( tRef, tauF, k ) {
  var tau = audE1.currentTime - tRef;
  // tRef: the playback position of cross-fade initiation
  if ( tau <= 0.0 ) { // before and at the beginning of cross-fading
    audE1.volume = 1.0; // 1st audio content of max. volume
    audE2.volume = 0.0; // 2nd audio content at silence
  }
  else { // during cross-fading
    var fadingOutVol = v0( tau, k );
    var fadingInVol = vI( tau, tauF, k );
    if ( fadingInVol < 1.0 ) { audE2.volume = fadingInVol; }
    if ( fadingOutVol > 0.0 ) { audE1.volume = fadingOutVol; }
    else { // cross-fade ending
      audE1.volume = 0.0; audE1.pause();
      // 1st audio content at silence (paused)
      audE2.volume = 1.0;
      // 2nd audio content of maximum volume
      window.clearInterval( timerId );
    }
  }
}

function playAndCrossFade( tRef, tauF, rhoH, k ) {
  audE1.currentTime = 0.0; audE1.volume = 1.0;
  audE1.controls = true;
  audE2.currentTime = 0.0; audE2.volume = 0.0;
  audE2.controls = true;
  compCoeff( tauF, rhoH, k );
  audE1.play(); audE2.play();
  timerId = window.setInterval( setVols, 50, tRef, tauF, k );
}

function crossFadeOf5s() {
  // starts playback and performs a 5 s cross-fade,
  // initiated at the playback position of 20 s
  butte.disabled = true;
  playAndCrossFade( 20.0, 5.0, 0.7, 4 )
  // passes the arguments: tRef = 20 s, tauF = 5 s, rhoH = 0.7, k = 4
}

```

```
function createElem( src1, src2 ) {
  butte = document.createElement( "BUTTON" );
  // creates the button element for allowing playback
  var butteTxt = document.createTextNode( "Play & cross-fade" );
  butte.appendChild( butteTxt );
  butte.addEventListener( "click", crossFadeOf5s );
  document.body.appendChild( butte );
  // appends the button to the document body

  var pE1 = document.createElement( "P" );
  var pE1Txt = document.createTextNode( "1st audio content" );
  pE1.appendChild( pE1Txt );
  document.body.appendChild( pE1 ); // 1st paragraph
  audE1 = document.createElement( "AUDIO" );
  // creates the 1st audio element
  audE1.src = src1;
  document.body.appendChild( audE1 );
  // appends the 1st audio element to the document body

  var pE2 = document.createElement( "P" );
  var pE2Txt = document.createTextNode( "2nd audio content" );
  pE2.appendChild( pE2Txt );
  document.body.appendChild( pE2 ); // 2nd paragraph
  audE2 = document.createElement( "AUDIO" );
  // creates the 2nd audio element
  audE2.src = src2;
  document.body.appendChild( audE2 );
  // appends the 2nd audio element to the document body
}

createElem( "sample1.mp3", "sample2.mp3" );
// cross-fade between the audio contents
// of the two mpeg files, provided by user

</script>
</body>
</html>
```

One observes that the values of the three coefficients interfering in the rational fractions (1), (3) result by calling the function “compCoeff()” and are computed according to relations (10) for the case of $v_m = 1$, which is highlighted by the transitions illustrated in Fig. 1 up to Fig. 4. As aforementioned, in this situation, quantity ρ_h in (9), denominated within the code by means of variable (parameter) “rhoH”, represents precisely the audio contents volume imposed at the cross-fade halfway point.

In order to receive the outputs of mappings (1) and (3) for a certain playback position within the cross-fade, functions “vO()” and “vI()” are invoked. In addition to parameters of function “vO()” that is the implementation of (1), function “vI()” includes the parameter “tauF” i.e. the cross-fade length. This is because function “vI()” represents just the implementation of the composite function (2), used here to fade the second audio content in. Both function “vO()” and function “vI()” are invoked inside the code of function “setVols()”, which is designed to automate the volume of each audio content. When it is called, function “setVols()” sets the volume property of the two audio objects, namely “audE1” and “audE2”, which point to the appropriate audio contents. Before the cross-fading, the volume of the first audio content, that has to be faded out, is preserved at the value of 1, which represents both the highest value adopted in HTML5 for the audio volume [13], [14], and the highest value of the volume of transitions depicted in Fig. 1 up to Fig. 4. During cross-fading, the “volume” property of audio object “audE1”, associated with the audio content to be faded out, is set at the return value of function “vO()” while the “volume” property of audio object “audE2”, associated with the audio content to be faded in, is set at the return value of function “vI()”.

It has to be emphasized that the implementation is optimized so that not only the playback position corresponding to cross-fade initiation but also the cross-fade attributes (length, shape of fade-out and fade-in transitions) can easily be managed by means of the arguments (parameters values) passed to function “playAndCrossFade()” only.

4. Conclusion

Both the fade-out and fade-in audio effects, required for carrying out audio cross-fades, are usually implemented in the off-line mode by means of various transcendental functions, such as logarithm, sine, and exponential function. Nevertheless, from the perspective of real-time computing, the receiving of both the volume of the audio section to be faded out and the volume of the section to be faded in exclusively by valuating the outputs of different transcendental functions could be regarded as very time consuming.

Consequently, to ensure that, during the cross-fade effect, the audio volume of each of the two overlapping sections is returned without noticeable delays, we have adopted appropriate rational functions to implement the fade-out and the fade-in audio effects, with the mention that the input of both functions is represented by the playback position within the cross-fade effect whilst the output of one of the functions is just the volume of the faded out audio section, and the output of the other function is precisely the volume of the faded in section.

The two mappings that allow the audio volume controlling have been chosen so that the volume at cross-fade midpoint of the faded out section is equal to the volume at midpoint of the faded in audio section. Hence, the cross-fade effect can straightforwardly be customized just by setting the ratio of the volume of the section to be faded out, occurring at midpoint, to the initial volume of the same audio section, i.e., practically, the maximum volume of each of the two separate audio sections.

The high level of versatility of the resulted cross-fade effect is pointed up by the transitions illustrated in Fig. 1 up to Fig. 4, where, in contrast to the cross-fades obtained for $\rho_h = 0.5$ and $\rho_h = 0.7$, the cross-fades corresponding to $\rho_h = 0.2$ introduce a volume dropping in the middle area of the effect, being, therefore, appropriate for the situation in which the tempi of the overlapped audio sections are exceedingly different. On the other hand, both the efficiency and convenience of customizing the cross-fade effect by means of the proposed rational functions are validated by a plain JavaScript implementation, with the discretization being achieved by employing the “setInterval()” method of the “window” object.

References

- [1] J. D. REISS, A. McPHERSON: *Audio Effects. Theory, Implementation and Application*. Boca Raton, FL, USA, CRC Press, 2015.
- [2] A. U. CASE: *Sound FX. Unlocking the Creative Potential of Recording Studio Effects*. Burlington, MA, USA, Focal Press, 2007.
- [3] W. JACKSON: *Digital Audio Editing Fundamentals. Get Started with Digital Audio Development and Distribution*. Berkeley, CA, USA, Apress Media, 2015. doi: 10.1007/978-1-4842-1648-4
- [4] S. LANGFORD: *Digital Audio Editing. Correcting and Enhancing Audio in Pro Tools, Logic Pro, Cubase, and Studio One*. Burlington, MA, USA, Focal Press, 2014.
- [5] C. SCHRODER: *The Book of Audacity. Record, Edit, Mix, and Master with the Free Audio Editor*. San Francisco, CA, USA, No Starch Press, 2011.
- [6] Audacity Manual. *Creating a Crossfade*, 2021. https://manual.audacityteam.org/man/creating_a_crossfade.html
- [7] V. COSTELLO: *Multimedia Foundations. Core Concepts for Digital Design*. New York, NY, USA, Routledge, 2016.
- [8] K.C.C. YANG, et al.: *Cases on Immersive Virtual Reality Techniques (Advances in Multimedia and Interactive Technologies)*. Hershey, PA, USA, IGI Global, 2019.
- [9] M. SWEET: *Writing Interactive Music for Video Games. A Composer’s Guide*. Upper Saddle River, NJ, USA, Addison-Wesley Professional, 2014.
- [10] F. CHEN, K. JOKINEN (Eds.): *Speech Technology. Theory and Applications*. New York, NY, USA, Springer Science + Business Media, 2010. doi: 10.1007/978-0-387-73819-2
- [11] J.P.H. VAN SANTEN, R. SPROAT, J. OLIVE, J. HIRSCHBERG (Eds.): *Progress in Speech Synthesis*. New York, NY, USA, Springer-Verlag, 1997. doi: 10.1007/978-1-4612-1894-4

- [12] T. DUTOIT: An Introduction to Text-to-Speech Synthesis. Dordrecht, Springer Netherlands, 1997. doi: 10.1007/978-94-011-5730-8
- [13] I. DEVLIN: HTML5 Multimedia. Develop and Design. Berkeley, CA, USA, Peachpit Press, 2012.
- [14] S. POWERS: HTML5 Media. Sebastopol, CA, USA, O'Reilly Media, 2011.
- [15] I. JACOBS, J. JAFFE, P. LE HEGARET: How the open web platform is transforming industry. *IEEE Internet Computing*, **16** (6), 82-86 (2012). doi: 10.1109/MIC.2012.134
- [16] L. LUPSA-TATARU: Audio fade-out profile shaping for interactive multimedia. *Technium: Romanian Journal of Applied Sciences and Technology*, **2** (7), 179-189 (2020). doi: 10.47577/technium.v2i7.1913
- [17] L. LUPSA-TATARU: Novel technique of customizing the audio fade-out shape. *Applied Computer Science*, **14** (3), 5-14 (2018). doi: 10.23743/acs-2018-17