Neural Network-based Violinist's Hand Animation

Junhwan Kim[1] Frederic Cordier[2] Nadia Magnenat-Thalmann[2]
[1]*Ecole Polytechnique* [2]*MIRALab, CUI, University of Geneva*
91128 Palaiseau France *24 rue du General Dufour, CH-1211 Geneva Switzerland*
junkim@poly.polytechnique.fr *{cordier,Nadia.Thalmann}@cuimail.unige.ch*

Abstract

We present a system for the animation of human hand that plays violin. Neural network controls the hand movement. We make use of an optimization method to generate the examples for the neural network training. The musical decision of which finger to use is automatically made by best first search. We will show that the movements of violinist's hands are physically and musically feasible, and that the musical decisions are consistent with those recommended in the violin pedagogy. A description of system, the results of the decisions, and the animations are presented.

1. Introduction

Visualization of human hand movement is a difficult problem. Until present, researchers have developed animation system of grasping motion [5,13], and sign language generation system [6]. In this paper, we present a system for the animation of human hand that plays violin. It poses several unprecedented challenges:

First, the movement of fingers and hands are more complicated than that of other activities, such as grasping. Thus we have to deliberately control the movements of each finger and hand to play a score. We make use of a set of *neural networks* (NN), where each NN is dedicated to each hand configuration.

Second, a musical score does not usually contain all the necessary information to play it. It does not specify every *fingering*; i.e., which finger to touch which string of violin. Thus it is up to the violinist to make such decisions. A series of researches have been focused on the fingering determination [9,14]. They make use of a set of rules inspired by the music pedagogy to decide fingering. Our system enjoys several advantages over such rule-based fingering determination methods: (1) the fingering can be determined according to different physical dimension or joint limits of the hands or fingers, as well as musical factors such as intensity and speed, (2) the method can be easily extended to the polyphonic score, whereas the number of rules must be unmanageably increased in the rule-based method.

We will show that the movements of violinist's hand are physically and musically feasible, and that the decisions on fingering coincide with or are similar to those recommended in the violin pedagogy. This paper is organized as follows: In section 2, we will present the related works. The system overview is given in section 3, the detailed explanations will be given in section 4 to section 9. We will present the results of musical decisions for the existing musical scores and some snapshots of the animations in section 10.

2. Related works

Since the seminal work of Ridsdale [12], neural network has been used as the one of the primary tool for motion control in the computer animation field. Panne and Fiume [8] developed "Sensor-Actuator Networks" for the physically-based animation of objects. The user supplies the configuration of a mechanical system, and then many possible modes of locomotion for the given objects are generated automatically by genetic algorithm. Sims [16] produced "Virtual Creatures" with neural network-based controller that generates motion signal according to sensory inputs. More recently, Grzeszczuk et al. [3] has shown that "NeuroAnimator" can replace physically based animation with marginal difference but orders of magnitude faster simulation time.

Until present, many researches have concentrated on the human hand animation. Rijpkema [13] and Huang et al. [5] developed knowledge-based grasp animation. Lee and Kunii [6] proposed a system that is capable of translating text from a natural language into animated sign language.

On the other hand, regarding on music performance animation Wood-Gain et al [19] developed a visualization system for drum playing. Users are allowed to change some parameters in the keyframes to affect the expression of the animation. Our system is different from theirs in (1) complexity of the movement, and (2) intelligent decision making. More recently, Sekiguchi and Eiho [15] realized the piano performance animation by hand and finger movements, which are constrained due to the anatomical components, such as tendons and bones. They also developed a fingering (i.e., assign an appropriate finger for each note) assignment algorithm.

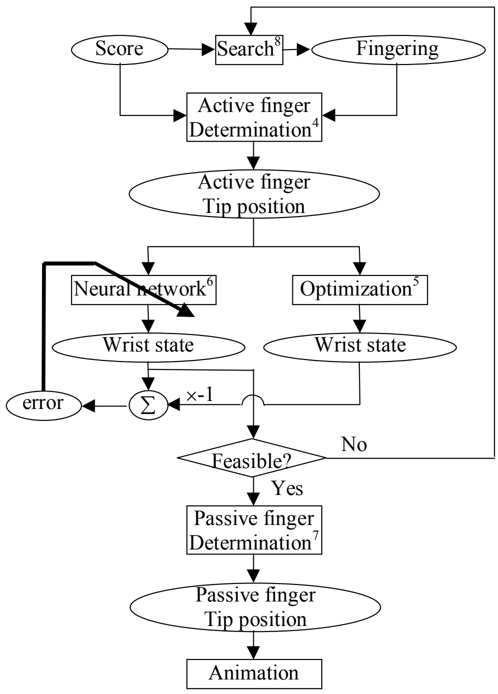

Figure 1. The system overview: Each module will be presented in the section indicated by superscript number.

3. System overview

The overall system is shown in Figure 1. For the given score, the *search* module suggests a *fingering*; i.e., a decision of which finger to use and which string to touch. According to the fingering and the score, *active finger determination* module generates the tip position of *active fingers*; i.e., fingers those are engaged in touch. Then *optimization* module generates the *wrist state*; i.e., position and orientation of wrist, which is used as examples for *neural network* training. It should be noticed that we make use of the optimization module only in training mode. By virtue of the wrist state, *passive finger determination* module calculates the joint angles of *passive fingers*; i.e., fingers those are not used in touch. If they are feasible, the animation module uses them to show the animation. Otherwise, the fingering determination module suggests another candidate fingering with the help of the previous results. We will describe each module in the following sections.

4. Active finger determination

This module determines the state of each finger and the tip position of the active fingers for each time unit. A score contains start time, end time, and intensity for each string touches. A string touch is a sequence of fingertip positions during a certain time period (see Figure 2). The height of the fingertip is plotted against the time axis. For a given touch, start time and end time of each state are determined by a couple of heuristics as follows:

The finger in idle state is triggered to pre-fly state if there remains just enough time to get to the desired position. In pre-fly state, the fingertip assumes a parabolic trajectory to the touch position. The end time of pre-fly state is to be start time of the given touch, which is also the start time of push state. The duration of push state and that of pull state are predefined to be same and short. We assume that there is no slip. Thus in push state, the fingertip goes straight down, by contrast in pull state, the fingertip goes straight up. In sound-generate state, the fingertip does not move. At the end of push state, there comes post-fly state whose duration is predefined to be rather short. The fingertip goes straight up in post-fly state. When the post-fly state ends, the finger returns to idle state.

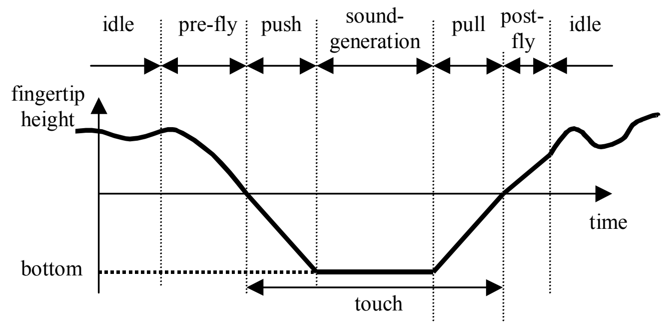

Figure 2. Height of fingertip during a string touch

5. Optimization

This module determines the wrist state given the fingertip position of active fingers computed in the previous module. We solve the problem by the optimization process. The evaluation function to be minimized depends on ease of execution, sound quality, and collision avoidance.

5.1. The evaluation function

5.1.1. Ease of execution. The movement of wrist position along the fingerboard is penalized following the rule of remaining the same position [14]. This rule is also relevant to sound quality, because by remaining in the same position, one has sections of string of approximately the same length that are vibrating, so the relative damping of the higher harmonics of the different notes is similar, thus producing a sound of uniform quality [14]. A cross between fingers is also penalized.

5.1.2. Sound quality. We adopt simple heuristic that the quality of sound is the function of the orientation of the fingertip that is in touch. It encourages the fingers to stay in *frictional cone* to prevent from sliding.

5.1.3. Collision avoidance. There are two possible collisions, that is, the collision between fingers and that between violin and a finger. We give penalties for passive fingers whose tip position is close to another finger or the violin.

5.2. Optimization procedure

Let $\mathbf{X_C}$ be the fingertip positions of active fingers, $\Theta_\mathbf{C}$ and Θ_R be the joint angles of active and passive fingers respectively, and $\mathbf{S}_W$ the wrist state. For given tip positions of active fingers or $\mathbf{X_C}$, we will find the wrist state or $\mathbf{S}_W$, and the angles of fingers ($\Theta_\mathbf{C}, \Theta_R$) such that the evaluation function $G(\Theta_\mathbf{C}, \Theta_R, \mathbf{S}_W)$ is minimized. If $\mathbf{S}_W$ is given, the joint angles of active fingers, or $\Theta_\mathbf{C}$ can be uniquely determined from $\mathbf{X_C}$ by *inverse kinematics* using constraints between the joint angles [13]. The *Newton-Rhapson method* [10] is employed for the calculation. Thus we have

$$\Theta_\mathbf{C} = K^{-1}(\mathbf{X}_C, \mathbf{S}_W)$$

where $K^{-1}(\bullet,\bullet)$ means inverse kinematics function to evaluate $\Theta_\mathbf{C}$ for given $\mathbf{X_C}$ and $\mathbf{S}_W$. Then $\Theta_R, \mathbf{S}_W$ can be determined by

$$\begin{aligned}(\Theta_R, \mathbf{S}_W) &= \arg\min_{\Theta_R, \mathbf{S}_W} G(\Theta_\mathbf{C}, \Theta_R, \mathbf{S}_W) \\ &= \arg\min_{\Theta_R, \mathbf{S}_W} G\left(K^{-1}(\mathbf{X}_C, \mathbf{S}_W), \Theta_R, \mathbf{S}_W\right)\end{aligned}$$

To put it differently, Θ_R and $\mathbf{S}_W$ are determined such that $G\left(K^{-1}(\mathbf{X}_C, \mathbf{S}_W), \Theta_R, \mathbf{S}_W\right)$ is minimized for given $\mathbf{X_C}$. We use *direction set method* [10] to find $\Theta_R, \mathbf{S}_W$ with the previous value as the initial value.

6. Neural Network

6.1. Structure

The input and the output of neural network (NN) are same as the optimization module; NN determines the wrist state given the fingertip position of active fingers. Each NNs can be categorized into 2 sorts (see Figure 3):

(1) *1 finger-NN* is employed when there is only one active finger. The input is the tip position of the active finger. The input enters the two NNs; *position-NN* and *orientation-NN*. The output of position-NN is the wrist position. It has no hidden layer (3-3 structure). On the other hand, the output of orientation-NN is the wrist orientation. It has one hidden layer with 6 nodes (3-6-3 structure) (see Figure 4).

(2) In case that there are two or more active fingers, *2 finger-NN* is used. We assume that cross between fingers does not occur. The input is the two extreme active fingers since the positions of *in-between* active fingers do not significantly affect the wrist state. The structure is similar to that of 1 finger-NN except that the dimension of input is 6. Thus the position-NN has 6-3 structure, whereas the orientation-NN has 6-6-3 structure (see Figure 4).

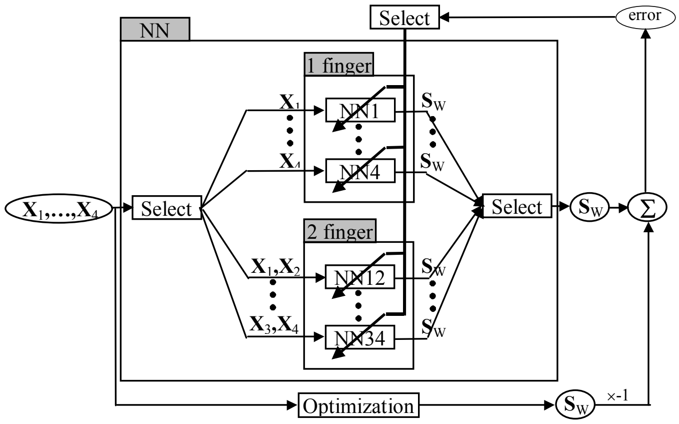

Figure 3. The structure of the neural network: NN*i* signifies the 1 finger-NN for active finger *i*, whereas NN*ij* denotes the 2 finger-NN for finger *i* and finger *j* as two extreme active fingers.

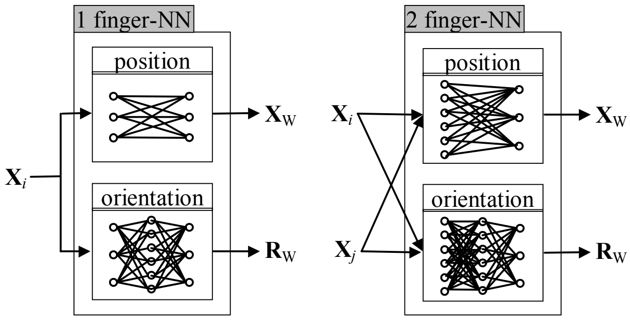

Figure 4. The structure of the neural network

6.2. Training and testing

Extensive experiment with different σ (learning coefficient) and *activation function* is beyond the scope of this work. However, we find that combination of σ=0.04 and *binary sigmoid function* [2] shows quite satisfying results.

6.3. Discussion

At first, we have performed experiment with one single NN – it has one or two hidden layers, the input consists of the tip position of active fingers (a predefined value is assigned for passive fingers) and 4 boolean variables indicating the finger is active or passive, and the output is wrist state. Unfortunately it does not show satisfying results. It is well known that an arbitrary function can be approximated by an NN with arbitrary precision [2], however the necessary number of hidden layer and nodes can be unacceptably large. We conclude that in our system, the necessary number of the hidden is quite large in case that we employ one or two hidden layers.

7. Passive finger determination

We make use of the same optimization method as in the active finger determination. However, the crucial difference is that (1) the dimension of search space is diminished by six (= DOF of the wrist state), (2) the search space is *separable*, thus dimension of search space is maximally 3. However, at first we make use of the *spring-damper model* [11,17]. Only in case that there is collision with other fingers or violin, we perform the optimization, which substantially expedites the calculation.

8. Search for fingering

The input to the search algorithm is a score without fingering. In a score, the attributes for each string touch include start time, end time, and intensity. A fingering, which generates feasible movements for hand and fingers to play a score, is determined based on all these attributes. An evaluation function is needed to estimate the amount of effort for the fingering. The goal of the search is to find such a fingering that makes the sum of the evaluation function sufficiently low, if not lowest.

Our strategy is a *best-first search* [18] as in [4]. Each node in the search tree represents a specific fingering of the same length as the depth of the node. As for polyphonic score, we "serialize" it, that is, we regard it as monophonic score from the lowest note to the higher one.

9. Bowing

The string player has access to four bowing parameters [1]: (1) *bow position*: the position of the contact point between bow and string, (2) *bow velocity*: the velocity of the bow transverse to the strings, (3) *bow force*: the force between bow and string, (4) *bow-bridge distance*: the distance along the string from the bridge to the contact point with the bow, (5) *bow-tilting angle*: the angle of bow tilting, which is used to reduce the contact width of the bow hair. Physically and musically feasible mapping from the score to these parameters is beyond the scope of this work. Instead, we adopt a simple heuristic: (1) entire portion of the bow is used in every slur except in the case that the velocity of the bow exceeds the predefined threshold, (2) up bows and down bows are used alternatively. Thus only the bow position and the bow velocity are employed to express the given score. It is justified by the fact that the bow-bridge distance and the bow-tilting angle are hardly noticeable and that this work is essentially kinematics-based so the bow force is not relevant.

Table 1. Fingering determination for two scores: G, D, A, and E represents four strings in violin and O, 1, 2, 3, and 4 denotes open, index finger, middle finger, pinky finger, and ring finger respectively. Piece 1 is the beginning of the G Major broken scale, whereas piece 2 is the beginning of the G Major 3rd scale

		sequence							
Piece1	Note	G4	G5	A4	A5	B4	B5	C4	C5
	string	G	D	G	D	G	D	G	D
	finger	O	3	1	4	1	4	1	4
Piece1 (cont.)	Note	D4	D5	E4	E5	f#4	f#5	G5	G6
	String	D	A	D	A	D	A	D	A
	Finger	O	3	1	4	1	4	1	4
Piece2	Note	G4	A4	B4	D5	C5	E5	D5	f#5
	string	G	G	G	D	G	D	G	D
	finger	O	1	2	O	3	1	4	2
Piece2 (cont.)	Note	E5	G5	f#5	A5	G5	B5	A5	C6
	String	G	D	G	D	G	D	G	D
	Finger	3	1	4	2	3	1	4	2

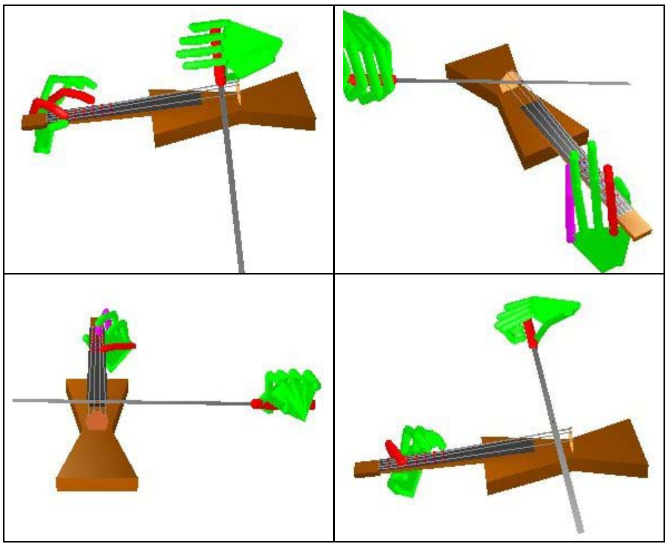

Figure 5. Snapshots of playing G Major broken scale

10. Results

10.1. Fingering determination

We have chosen two pieces from a standard technique book [7]. The fingering found by our method coincide with the recommended fingering (see Table 1).

10.2. Animation

Figure 5 shows some snap shots of the animation playing the piece 1.

11. Conclusions and Further works

We have described the neural network-based system for the animation of violinist's hand. The animation shows feasible movement of hands and fingers. In general, the result of the fingering determination accords with the written fingerings. The advantage of this fingering determination method over the rule-based method is adaptability to different factors, ability to consider global effect, and scalability to polyphony.

At present, we employed a set of neural networks, which requires extensive training for each sub neural networks. A well-designed one single neural network would have advantage of shorter training time and simpler structure over this set of neural networks. We will develop and test such neural network in near future.

Although we have made use of neural network only for the calculation of wrist movement, it would be desirable to develop a neural network for the calculation of finger joint angles as well. The hand movement generated by our system is purely *neutral*; i.e., it does not indicate an emotional state, or a musical character. A music cognition model can be a solution.

12. References

[1] A. Askenfelt, "Measurement of the bowing parameters in violin playing. II: Bow-bridge distance, dynamic range, and limits of bow force," *Journal of the Acoustical Society of America*, 91(6):3540-3550, June 1992.

[2] L. Fausett, *Fundamentals of Neural Networks*, Prentice Hall International, Inc. 1994.

[3] R. Grzeszczuk, D. Terzopouos, G. Hinton, "NeuroAnimator: Fast Neural Network Emulation and Control of Physics-Based Models, " *Proceedings of SIGGRAPH* '98, pp 9-20, 1998.

[4] P. S. Huang and M. van de Panne. "A Search Algorithm for Planning Dynamic Motions," *Computer Animation and Simulation '96 Proceedings of the 7th Eurographics Workshop on Simulation and Animation*, Springer Verlag, pp. 169-182.

[5] Z. Huang, R. Boulic, N. Magnenat Thalmann, D. Thalmann, "A Multi-sensor Approach for Grasping and 3D Interaction," *Proceedings of Computer Graphics International* `95, pp.235-254.

[6] J. Lee, and T. L. Kunii, "Computer Animated Visual Translation From Natural Language to Sign Language, " *The Journal of Visualization and Computer Animation*, Vol. 4: 63-78 (1993).

[7] E. Nerini, *Gammes et Arpeges pour le Violon, 3^{e} Cahier: Gammes en tierces, en sixtes et en octaves*, Editions M. Combre.

[8] M. van de Panne and E. Fiume, "Sensor-Actuator Networks, " *Proceedings of SIGGRAPH* '93, pp 335-342, 1993.

[9] R. Parncutt, J. A. Sloboda, E. F. Clarke, M. Raekallio, and P. Desain, "An Ergonomic Model of Keyboard Fingering for Melodic Fragments," *Music Perception*, 4: 341-381, 1997.

[10] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, The art of scientific computing, Cambridge University Press, 1988.

[11] M. H. Raibert and J. K. Hodgins, "Animation of dynamic legged locomotion," *Proceedings of SIGGRAPH* '91, pp. 349-358.

[12] G. Ridsdale, "Connectionist Modeling of Skill Dynamics, " *The Journal of Visualization and Computer Animation*, 1: 66-72, 1990.

[13] H. Rijpkema, "Computer animation of knowledge-based human grasping," *Proceedings of SIGGRAPH* '91, pp. 339-348.

[14] S. I. Sayegh, "Fingering for string instruments with the optimum path paradigm," *Computer Music Journal*, 13 (3): 76-83, 1989.

[15] H. Sekiguchi and S. Eiho, "Generating and Displaying the Human Piano Performance, " *Transactions of Information Processing Society of Japan*, 40(6) :2827-2837, 1999. (*in Japanese*)

[16] K. Sims, "Evolving Virtual Creatures," *Proceedings of SIGGRAPH* '94, pp. 15-22.

[17] N. Torkos and M. van de Panne, "Footprint-based Quadruped Motion Synthesis," *Proceedings of Graphics Interface '98*, pp. 151-160.

[18] P. H. Winston, *Artificial Intelligence*, 2nd Edition, Addison-Wesley, 1984.

[19] A. Wood-Gaines, F. D. Fracchia, and T. Calvert, "Visualizing Expressive Movement in Music, " *IEEE Visualization '96, Late Breaking Hot Topics*, 1996.