Development of a Piano-Playing Robot with Motion-Sound Mapping

Thai V. Hoang*, Kwang-Hyun Park**, and Zeungnam Bien***

* Department of Industrial Automation, Hanoi University of Technology
  No 1 Dai Co Viet, Hanoi, Vietnam
  e-mail: hvthaibk@gmail.com
** Department of Information and Control Engineering, Kwangwoon University
  447-1 Wolgye-dong, Nowon-gu, Seoul, 139-701, Republic of Korea
  e-mail: akail@kw.ac.kr
*** Department of Electrical Engineering and Computer Science, KAIST
  373-1 Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea
  e-mail: zbien@ee.kaist.ac.kr

Abstract — This paper presents the development of a piano-playing robot with motion-sound mapping in order to provide the elderly a means of entertainment that can interact with them while it is in operation.

A piano-playing robot composed of a personal computer, a robotic arm, and a robotic hand is first developed to make a system that can play piano mechanically. Kinematic, dynamic equations, and Jacobian matrix for the robotic arm and the robotic finger are then derived for real-time control and simulation of the system. A study on the actions underlying sound generation in acoustic piano is carried out to have an insight into the relations between piano sound and keys' motion. These relations are later represented by neural networks. By using these relations, the piano-playing robot can produce fascinating music as a result of mimicking sample music.

Experiment results demonstrate that the developed system outperforms existing systems in the sense of music attractiveness and make it a candidate to be used as a service robot in entertaining human being.

Index Terms — Piano-Playing Robot, Motion-Sound Mapping, Affective Interaction

1. INTRODUCTION

As the number of the elderly is steadily increasing in our society, the demand for medical care and welfare services also increases. Human-friendly man-machine interaction systems are thus getting more and more attention as a means of care-giving aids, such as intelligent robotic systems that are capable of harmonious coexistence with human beings.

Some researchers focus their work on building smart houses with automation systems where almost all simple and repetitive tasks are executed by computers or microprocessors. These kinds of systems can be easily realized in almost all kinds of appliances, alarm systems, and can help the elderly to live independently. However, these advanced systems are purely physical; there is no connection between these systems and the emotional states of the people living in the house. Thus, affective states of the elderly staying alone in the house may be missed and this can lead to their affective disorder.

Recent years have seen an increasing number of researchers focusing their work on building intelligent houses [2][10]. In addition to being equipped with conventional automation systems, these houses can also interact with their inhabitants to provide a more comfortable living environment.

In order to build a house as friendly as possible, a survey was conducted to determine how to best assist the elderly in their daily life. The survey results shown in Figure 1 reveal that a large part of the elderly (25%) prefers singing. Thus, to enhance mental care of the elderly, a piano-playing robot could be a solution which provides people with hours of healthy entertainment. The tempo, tone, etc. of the music played by the robot
can be adjusted according to the users’ emotional expression by gesture.

Figure 1. The elderly hobby survey results

With the current state of the art, it is impossible to make a piano-playing robot that would match the human in playing piano. Some systems focus on the sophisticated mechanical design of anthropomorphic hand and arm but some just pay attention to the reproduction of the music itself, regardless of their structure. For example, the WABOT-2 robot [12] and the DLR Hand II [1] both have anthropomorphic appearances due to complex mechanical designs but use relatively simple control strategies in controlling the robot to play piano. The WABOT-2 robot uses point-to-point control for robotic fingertip whereas the DLR Hand II tries to track the robotic fingertip to follow an identical trajectory, which is obtained by moving the robotic fingertip manually during the teaching stage, in striking each piano key. Consequently, the music produced by these robots is unattractive.

On the other hand, the Automatic Piano [5] utilizes a complex control structure and thus guarantees to produce high quality music but still imposes some limitations. It does not have human-friendly appearance and needs plays of a professional pianist for the creation of a trajectory database, which does not always available for every piece of music.

A new piano-playing robot system that both has sophisticated mechanical designs and high performance control algorithms is, therefore, needed to be developed. An overview of the piano-playing robot system that has interaction with the elderly is illustrated in Figure 2. We can easily recognize from this figure that there are three main different tasks to be completed in building a piano-playing robot system: reading music scores, controlling the robot to strike piano keys, and detecting human gestures.

Figure 2. A piano playing robot system

The tasks of reading music scores and detecting human gestures have been accomplished in [9] an [7] respectively. Thus, this paper is devoted to the development of a robot that not only can play piano but also outperform the current systems in the sense of music attractiveness. The general idea behind the described task is to develop a piano-playing robot mechanically and make it to play piano as attractive as possible.

The underlying questions for the development of a piano-playing robot are “How the robot is designed mechanically to be able to play piano, or to strike piano keys?” and “How to build a control system for this robot and control it so it can operate as designed?”. Details of the development of a piano-playing robot will be presented in section 2.

For the problem of making the robot to play music attractively, the solution is to equip the robot an ability of learning to play piano like that of human being. Actually, there are two distinguished approaches for human to learn to play piano:

- The first one is called trajectory tracking. In this approach, students try to move their hands in the same trajectories as that of teacher's hands.

- The second approach comes from the idea of mimicking the music directly. In this way, students try to self-adjust the movements of their hands in order to produce music as similar as the sample one.

For real application, the first approach imposes some limitations because of a large number of trajectories needed to follow in one piece of music and the inconsistency of trajectories generated by
human hands even for one note with a particular pitch between different strikes. Thus, the approach of mimicking the sound directly will be used for the piano playing robot in learning to produce attractive music. Detailed information related to the foundation and implementation of this method will be presented in section 3.

2. DEVELOPMENT OF A PIANO PLAYING ROBOT

2.1 Hardware Structure of the Piano-Playing Robot System

With the goal of making a piano-playing robot that has an anthropomorphic appearance and has the same functionality with a human arm and a human hand in playing piano, the piano-playing robot system should have a robotic arm with 6 degrees of freedom and a robotic hand composed of 5 fingers, each finger has 3 degrees of freedoms. The required components for a piano-playing robot system are sketched in Figure 3. The central controller is a Pentium III 850Mhz personal computer. The piano used is of the model Kurzweil RE-210. The robotic arm is built from PowerCube modules, among which are 4 rotary units and one wrist unit, to have a robot with 6 degrees of freedom of the same kinematic structure with the Puma560 robot. The robotic hand which meets the kinematic requirements has been designed and made in Bien’s System Control Lab (KAIST).

![Figure 3. Components of a piano playing robot system](image)

Figure 4 provides a connection scheme that is used in controlling the piano-playing robot. The data bus used for exchanging data is a CAN bus. Each controller is used to control one robotic finger which uses 2 DC motors and one stepper motor as actuators.

![Figure 4. Connection scheme of the control system for the robotic hand](image)

As the robotic hand is used for playing piano so the control of robotic fingers requires high performance. Thus, one digital signal processor (Microchip dsPIC30F4012) is employed to design one controller. Inside the digital signal processor, control algorithms used to control actuator, PID control, is implemented. This can be a position or velocity controller based on the required control parameter. Figures 5 provides a block diagram of the controller.

![Figure 5. A block diagram for position/velocity control of a DC motor](image)

At a point of time, only one type of controller, position or velocity, is utilized as the controller for the DC motor:

- Position controller is employed when the robotic fingers release piano keys and prepare for the next strikes.
- Meanwhile, velocity controller is used to control the robotic fingers to strike piano keys. In this stage, the fingertip velocity should be at a constant value. This is due to analysis results on the actions underlying sound generation in pianos that will be
presented in section 3.

A picture of the implemented system that follows the above features and design philosophy is shown in Figure 6.

![Figure 6. The implemented piano-playing robot system](image)

2.2. Robotic Finger Modelling

2.2.1 Forward Kinematic Equations

Each finger of the robotic hand has four rotational joints with the last two joints are actuated by only one motor. The kinematic configuration of one finger is shown in Figure 7, where frame $O_0X_0Y_0Z_0$ and $O_iX_iY_iZ_i$ are the base and fingertip coordinate systems respectively. Table 1 provides the link parameters for the robotic finger.

![Figure 7. Kinematic configuration of one robotic finger](image)

Using the method in [4], the transformation that relates the fingertip to the base is given below:

$$
\mathbf{T}_{ip} = \begin{bmatrix}
  r_{11} & r_{12} & r_{13} & p_x \\
  r_{21} & r_{22} & r_{23} & p_y \\
  r_{31} & r_{32} & r_{33} & p_z \\
  0 & 0 & 0 & 1
\end{bmatrix}
$$

with

$$
\begin{align*}
  r_{11} &= s_1c_2s_34 + c_1c_34 \\
  r_{12} &= s_1s_2 \\
  r_{13} &= s_1c_2c_34 + c_1s_34 \\
  p_x &= s_1c_2a_2 + (s_1c_2c_3 + c_1s_3) a_3 + (s_1c_2c_34 + c_1s_34) a_4 \\
  r_{21} &= s_2s_34 \\
  r_{22} &= c_2 \\
  r_{23} &= -s_9c_34
\end{align*}
$$

### Table 1. Kinetic parameters of one robotic finger

<table>
<thead>
<tr>
<th>link</th>
<th>$\theta_i$</th>
<th>$a_{i-1}$</th>
<th>$d_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\theta_1 - 90^\circ$</td>
<td>-90°</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$\theta_2$</td>
<td>-90°</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>$\theta_3$</td>
<td>90°</td>
<td>$a_2$</td>
</tr>
<tr>
<td>4</td>
<td>$\theta_4$</td>
<td>0°</td>
<td>$a_3$</td>
</tr>
</tbody>
</table>

$$
\begin{align*}
  p_y &= -s_2a_2 - s_2c_3a_3 - s_2c_3a_4 \\
  r_{31} &= -c_1c_2c_34 - s_1c_34 \\
  r_{32} &= c_1s_2 \\
  r_{33} &= c_1c_2c_34 - s_1s_34 \\
  p_z &= c_1c_2a_2 + (c_1c_2c_3 - s_1s_3) a_3 + (c_1c_2c_34 - s_1s_34) a_4
\end{align*}
$$

where $c_i = \cos \theta_i$, $s_i = \sin \theta_i$.

2.2.2 Inverse Kinematic Equations

Inverse kinematic equations for one robotic finger can be directly obtained from its geometric configuration. Suppose the robotic finger has its geometric configuration as in Figure 8, the joint angle $\theta_2$ can be calculated as follows:
\[ \theta_2 = \arctan \frac{2(x_t, y_t)}{z_t} \]  

For the joint angles \( \theta_3 \):

\[ d = 4a_2a_4 \]
\[ e = 2a_2s_3 + 2a_3a_4 \]
\[ f = a_2^2 + a_3^2 + x_t^2 - y_t^2 - z_t^2 - 2a_2a_4 \]
\[ \Rightarrow d \cos^2 \theta_3 + e \cos \theta_3 + f = 0 \]  

Figure 8. Geometric configuration for inverse kinematic calculation

\[ \Rightarrow \theta_3 = \arccos \frac{-e \pm \sqrt{e^2 - 4df}}{2d} \]  

For the joint angles \( \theta_1 \):

\[ g = a_2 + a_3 \cos \theta_3 + a_4 \cos(2\theta_3) \]
\[ h = a_3 \sin \theta_3 + a_4 \sin(2\theta_3) \]
\[ \Rightarrow g \sin \theta_1 + h \cos \theta_1 = z_t \]
\[ \Rightarrow \theta_1 = \arcsin \left( \frac{z_t - \sqrt{z_t^2 - g^2 - (g^2 + h^2)(z_t^2 - h^2)}}{g^2 + h^2} \right) \]  

2.2.3 Jacobian

From the equations obtained in the previous subsection, the Jacobian matrix of the robotic finger can be developed following a method provided in [4] and is given in the form below:

\[ J = \begin{bmatrix}
  j_{11} & j_{12} & j_{13} & j_{14} \\
  j_{21} & j_{22} & j_{23} & j_{24} \\
  j_{31} & j_{32} & j_{33} & j_{34}
\end{bmatrix} \]  

The above is a 3 \( \times \) 4 Jacobian matrix. This is because the robotic finger has 4 joints but only 3 degrees of freedom. This matrix cannot be used as a transformation between the joint velocities and the fingertip linear velocities. However, because of the coupling between third and fourth joints, \( \theta_3 \approx \theta_4 \), a new Jacobian matrix can be deduced from (5) by adding the third and fourth columns:

\[ \tilde{J} = \begin{bmatrix}
  \tilde{j}_{11} & \tilde{j}_{12} & \tilde{j}_{13} \\
  \tilde{j}_{21} & \tilde{j}_{22} & \tilde{j}_{23} \\
  \tilde{j}_{31} & \tilde{j}_{32} & \tilde{j}_{33}
\end{bmatrix} \]  

with

\[ \tilde{j}_{11} = c_2(a_2c_{34} + a_3c_4 + a_4) \]
\[ \tilde{j}_{12} = 0 \]
\[ \tilde{j}_{13} = a_3c_4 + 2a_4 \]
\[ \tilde{j}_{21} = -s_2(a_3s_3 + a_4s_{34}) \]
\[ \tilde{j}_{22} = -(a_2 + a_3c_3 + a_4c_{34}) \]
\[ \tilde{j}_{23} = 0 \]
\[ \tilde{j}_{31} = c_2(a_2s_{34} + a_3s_4) \]
\[ \tilde{j}_{32} = 0 \]
\[ \tilde{j}_{33} = a_3s_4 \]

The Jacobian matrix in (6) is employed for velocity control of the robotic fingertips.

2.2.4 Dynamic Equations

The dynamic system of one finger can be divided into two parts. One is the dynamics of actuating motor and transmission system, the other is the dynamics of the finger itself. The dynamic equations for one robotic finger has been developed using the Iterative Newton-
Euler algorithm [4]. The symbolic form of dynamic equations are as follows:

$$\tau_L = M_L(\theta)\ddot{\theta} + V_L(\theta, \dot{\theta}) + G(\theta) + F_L(\theta, \text{sign}(\dot{\theta}))$$

(7)

where:

- $\tau_L$ is the driving torque
- $M_L(\theta)$ is the $3 \times 3$ mass matrix of the finger
- $V_L(\theta, \dot{\theta})$ is the $3 \times 1$ vector of centrifugal and Coriolis terms
- $G(\theta)$ is the $3 \times 1$ vector of gravity term
- $F_L(\theta, \text{sign}(\dot{\theta}))$ is a $3 \times 1$ viscous and Coulomb friction
- $\theta$ is the $3 \times 1$ vector of the joint angles

The detailed expansion of (7) is given below:

$$\tau_1 = m_{11}\ddot{\theta}_1 + m_{12}\ddot{\theta}_2 + m_{13}\ddot{\theta}_3 + b_{112}\dot{\theta}_1\dot{\theta}_2 + b_{113}\dot{\theta}_1\dot{\theta}_3 + b_{123}\dot{\theta}_2\dot{\theta}_3 + c_{12}\dot{\theta}_2^2 + c_{13}\dot{\theta}_3^2 + k_1g$$

(8)

$$\tau_2 = m_{21}\ddot{\theta}_1 + m_{22}\ddot{\theta}_2 + m_{23}\ddot{\theta}_3 + b_{212}\dot{\theta}_1\dot{\theta}_2 + b_{213}\dot{\theta}_1\dot{\theta}_3 + b_{223}\dot{\theta}_2\dot{\theta}_3 + c_{22}\dot{\theta}_2^2 + c_{23}\dot{\theta}_3^2 + k_2g$$

(9)

$$\tau_3 = m_{31}\ddot{\theta}_1 + m_{33}\ddot{\theta}_3 + b_{312}\dot{\theta}_1\dot{\theta}_2 + b_{313}\dot{\theta}_1\dot{\theta}_3 + c_{31}\dot{\theta}_1^2 + c_{32}\dot{\theta}_2^2 + k_3g$$

(10)

The dynamics of a DC motor and its transmission system can be described as follows:

$$\tau_m = J_m\ddot{\theta}_m + B_m\dot{\theta}_m + f_m + \frac{\tau_L}{n}$$

(11)

where

- $\tau_m = 3 \times 1$ applied motor torques
- $\tau_L = 3 \times 1$ link joint torques
- $J_m = 3 \times 1$ motor’s inertia
- $B_m = 3 \times 1$ motor’s damping constants
- $f_m = 3 \times 1$ motor’s frictions
- $n = 3 \times 1$ reduction ratio
- $\theta_m = 3 \times 1$ motor’s positions

Combining (7) and (11) we have the following relation between the input $\tau_{\text{act}}$ and the output $\theta$:

$$\tau_{\text{act}} = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) + F(\theta, \text{sign}(\dot{\theta})) + J^T F_{\text{ext}}$$

(12)

where

$$\tau_{\text{act}} = \frac{\tau_m}{n}$$

$$M(\theta) = M_L(\theta) + \frac{1}{n^2}J_m$$

$$V(\theta, \dot{\theta}) = V_L(\theta, \dot{\theta}) + \frac{1}{n^2}B_m\dot{\theta}$$

$$F(\theta, \text{sign}(\dot{\theta})) = F_L(\theta, \text{sign}(\dot{\theta})) + \frac{1}{n}f_m$$

$$F_{\text{ext}} = 3 \times 1$$ external forces on the fingertip

Equation (12) is the dynamic equation for the robotic finger. It will then be used in to find the best PID parameters through simulation.

3. A NOVEL APPROACH BY SOUND MIMICKING

In this section, a sample music is first analyzed quantitatively to decide how the music can be mimicked. Figure 9 shows the oscillogram of the first part of the music Fur Elise. From the figure we realize similar shapes in oscillogram, the most prominent difference is how big those shapes are.

This observation triggers an idea of mimicking the music by trying to match its shape in its oscillogram. The following subsection on the ADSR envelope gives a background that supports this idea and makes it legitimate.

![Oscillogram of the first part of the sample Fur Elise](image)

3.1 The ADSR Envelope

When a mechanical instrument produces sound, the sound volume changes over time and the manner
of this change is different from instrument to instrument. For example, the sound of a piano has its highest volume immediately after it is pressed and fades gradually with time.

In electronic musical instruments, an envelope named ADSR is used to tailor a timbre in order to produce a sound more like that of a mechanical instrument. The ADSR envelope is characterized by using four parameters as seen in Figure 10:

![ADSR Diagram](image)

Figure 10. Schematic of ADSR

- **Attack (A):** how quickly the sound reaches its full volume after a note is activated (a key is pressed).
- **Decay (D):** how quickly the sound drops to the sustain level after the initial peak.
- **Sustain (S):** the constant volume that the sound takes after decaying until the note is released.
- **Release (R):** how quickly the sound fades when a note ends (a key is released).

For the case of a digital piano, there is no continuous steady level. The decay of sound happens after a key is depressed and holds; the fade of sound happens when a key is released. This means that the decay rate and the release time for one piano key depend only on the piano itself, not on the way the key is depressed or released. Thus, there is only one parameter of piano sound that can be a candidate for the mimicking process, the attack time.

We know that the longer the attack time is, the louder the sound will be. Therefore, the amplitude of sound will then be the only parameter that can be used for sound mimicking. The next subsection presents the relation between the way the piano key is depressed and the sound amplitude it produces.

### 3.2 Piano Actions

Figure 11 shows an action diagram of an upright piano. A sequence of actions underlying a key strike is described below.

When a piano key is depressed slowly, it rocks on the center rail and goes up in back. The key raises the sticker and wippen. The wippen pushes the jack, which pushes the hammer butt. The hammer butt pivots on its flange and moves the hammer toward the string. When the key is half way down, the spoon engages with the damper lever, lifting the damper off the strings. When the hammer is almost to the strings, the jack heel bumps into the regulating button, and as the wippen keeps going up, the jack pivots and slips out from under the hammer butt. The hammer continues under its own inertia to the string, instantly rebounding. At this point the strings start vibrating; the vibrations are carried to the bridge which transmits the vibration to the soundboard (the large, thin wood piece you can see in the back of the piano) which amplifies the sound (like a big speaker).

The catcher is caught by the back-check and held in this position as long as the key is depressed. When the key is released, the wippen drops, the back-check releases the catcher, and with the help of the butt spring, the hammer returns to the hammer rail. The damper spring returns the damper to the strings, and the jack spring returns the jack under the butt, ready for the next repetition. This entire sequence occurs in a fraction of a second, allowing the pianist to repeat notes rapidly.

The following observations are inferred from the above description and are confirmed in literature [8]:

- The sound produced by a piano has bigger amplitude if the vibration amplitude of the string is bigger.
- The vibration amplitude of the string depends on how hard the hammer strikes the string. The harder the strike is, the bigger the vibration amplitude will be.
How hard the hammer strikes the string depends on its momentum.
- The momentum of the hammer is constant from the moment it escapes from the jack.
- This constant momentum is a function of angular velocity of the hammer at the moment of escapement.
- The hammer’s angular velocity has a linear relationship with the angular velocity of the depressed piano key.

From the above observations we have the following conclusions:

- The sound produced by an acoustic piano only depends on the velocity of a piano key at the moment of escapement of the hammer from the jack.
- There exists a increasing monotone relation between the amplitude of the sound produced by an acoustic piano and the velocity of a piano key.

In the case of a digital piano, optical sensors are utilized to detect the keys’ velocity which in turn will be used as the inputs for the generation of sound. Thus, the above conclusions is still valid with digital pianos and they suggest an idea of constant velocity control of the robotic fingertip during a key strike.

However, at the current stage, if the piano playing robot is required to produce sound with a specific amplitude, it does not know at which velocity the fingertip should move.

By utilizing the increasing monotone relation between the amplitude of the sound produced by a piano and the velocity of a piano key, it would be very convenient if there is a mathematical relation between these two quantities. The next subsection is devoted to the building of that relation.

3.3 Motion-Sound Mapping Using Neural Networks

The relation between sound amplitude and key velocity depends on the piano itself and there is no concise mathematical representation for this relation. The only thing we can have is the sample values of sound amplitude, velocity obtained from experiment.

As stated in the previous subsection that this relation is increasing monotone, then a mapping from sound amplitude and fingertip velocity can be easily generalized from experiment samples using neural networks with one input and one output.

One problem arises here is the learning scheme for neural networks. In control area, there are two learning schemes [6] used for inverse model learning and plant model learning. However, they are not appropriate for learning mappings between sound amplitude and key velocity. Thus, a new learning scheme has been proposed and is shown in Figure 12.

In this learning scheme, the neural network input and output are the desired sound amplitude and the robotic key velocity. This velocity is then used as the reference value for the robotic fingertip velocity control to strike a piano key. The difference between the actual sound amplitude produced by piano and the desired sound amplitude is then used to train the neural networks by means of backpropagation algorithm.

The following settings are used for carrying out experiment to have sample values of amplitude, velocity:

![Neural network diagram](image)

**Figure 12.** Neural networks learning scheme in the piano playing robot

- A robotic finger is used to strike piano keys.
- The fingertip velocity for each strike is constant.
- Each piano key is struck with 16 different velocity values.
- Both the fingertip velocity and the sound produced are recorded.

Figure 13 shows the sampled values of fingertip velocity and sound amplitude obtained from experiment. The learning process is conducted in MATLAB with backpropagation algorithm using 20 neurons in the hidden layer. Figure 14 shows an obtained mapping for the key D5. The error between the actual and learned velocities is shown in Figure 15.
Figure 16 shows a block diagram of the velocity selection and control of a robotic finger for a specific desired sound volume and desired music note. The required velocity value is selected from a database of motion-sound mappings which has been built before.

3.4 Velocity Control of Robotic Finger

To complete the discussion on the issue of controlling robotic finger using motion-sound mappings, this section is devoted for high performance velocity control of robotic finger in Cartesian space.

The actuators used to drive robotic finger's links are DC motors with rated power far exceed requirements. In that sense, controllers for these motors are built upon PID rule for simplicity. Figure 17 shows the step response of a robotic finger joint with the selected $K_p = 10$, $K_i = 0$, $K_d = 1$.

At first sight, the step response in Figure 17 seems to be very good as it is fast and does not have overshoot. However, let’s have a look on the PWM control signal for the DC motor in Figure 18. The PWM value usually get saturated even when the robotic joint reaches its reference angle. This phenomena is not preferred as it may cause small vibration in the system. Therefore, there is a need to change the PID parameters to get a better performance.

But how can the PID parameters are changed? The next subsection will use a a search-based optimization method to obtain the best PID parameters.

By using the dynamic equations of the robotic finger built in subsection 2.2.4, PID parameters can be appropriately selected by means of a search-based optimization method through simulation.

One search-based optimization method that is widely used in simulation is the genetic algorithm [3]. In literature, genetic algorithm has long been used for optimal selection of PID parameters [13], [11]. The algorithm provided in [13] is repeated with the following conditions:

- The plant to be controlled is one robotic finger. In simulation, its dynamic model is used.
- The fitness function used for the evaluation process is the reciprocal of the integral of absolute magnitude of error $J_f = \frac{1}{\int t \cdot |e(t)| dt}$.

The value of fitness function during simulation is shown in Figure 19. The final PID parameters are as follows:

$$K_p = 30.31, K_i = 11, K_d = 0.72$$

Figure 20 shows the step response of a robotic finger joint with $K_p = 30.31$, $K_i = 11$, $K_d = 0.72$ and it demonstrate an improvement in velocity control of robotic finger. These PID parameters are currently used in the control of robotic finger.

4. Experiment Results

Figure 9 and 21 show the oscillograms of the first part of the sample Fur Elise (SFE) and the Fur Elise produced by the piano-playing robot (FER).

As SFE is produced using an acoustic piano, the shapes of sound in the oscillogram are quite strange and do not totally match the ADSR envelope like in the case of FER.

Another observation is that the time needed to play SFE is about 3.5 seconds while FER needs approximately 6.15s to be played. The reason for this slowness of FER is the limitations in mechanical design of the robotic hand that prevent its fingers from moving with high speed. As the decay rate of piano sound cannot be controlled, along the time axis there are some intervals that the sound amplitude almost fades to zero. This may makes the listener feels a pause in FER.

Except for those differences, the music generated by the piano-playing robot system quite match the sample music. Its attractiveness has been confirmed by human.

5. Conclusions

In this paper, a piano playing robot system has been developed and a new approach to mimic a sample music has been proposed.

The piano playing robot system is composed of a digital piano, a personal computer, a robotic arm, and a robotic hand. The robotic arm has a kinematic structure of the PUMA560 and has 6 DOFs. The robotic
hand has 5 fingers with 3 DOFs each. Controllers for these robotic arm and robotic fingers are connected to CAN buses and then connected to the personal computer for the exchange of reference and feedback data. Modeling of both robotic arm and robotic fingers have been carried out for the purpose of realtime control and simulation.

To cope with the problem of the attractiveness of music produced by the piano playing robot, we have first studied the actions underlying the sound generation in acoustic piano to realize that the piano sound is only influenced by the keys’ velocity at the moment of escapement of the hammer from the jack. With this information, we have proposed an approach of using the mappings between sound amplitude and keys’ velocity for the mimicking of the robot to sample music. In doing this, a new neural network learning scheme specific to motion-sound mapping has also been proposed.

High performance velocity control of the robotic finger in Cartesian space has been dealt with by PID control. Proper PID parameters selection has been done by the use of genetic algorithm in simulation.

Experiments have been carried out to demonstrate the working of the piano playing robot system and the effectiveness of the motion-sound mapping approach in producing attractive music.

REFERENCES


Figure 13. Experiment results for the piano key D5

Figure 14. A motion-sound mapping for the piano key D5

Figure 15. The error between actual and learned mapping

Figure 16. A block diagram of the velocity selection and control

Figure 17. Step response of a joint with $K_p=10$, $K_I=0$, $K_D=1$

Figure 18. PWM values over time with $K_p=10$, $K_I=0$, $K_D=1$

Figure 19. Fitness values during simulation
Figure 20. Step response of a joint with $K_p = 30.31$, $K_I = 11$, $K_D = 0.72$

Figure 21. Oscillogram of the first part of the Fur Elise produced by the piano playing robot