93-05
Van den Berghe G., Minten W., De Moor B., "Modeling a grand piano key action", *Computer Music Journal*, vol. 19, no. 2, 1995, pp. 15-22., Lirias number: 71913.

Guido Van den Berghe, Bart De Moor, and Willem Minten
Electrical Engineering Department
Katholieke Universiteit Leuven
Kard. Mercierlaan 94
3001 Leuven, Belgium
guido.vandenberghe@esat.kuleuven.ac.be

Modeling a Grand Piano Key Action

In this article we report on part of our research in piano action models, hoping to convey the need for research on this topic by explaining the shortcomings of most synthesizer keyboard-scan techniques. We have developed a new piano-key action model, based on a detailed description of a grand piano action. Every component plays a role in the specific behavior of the mechanism. Our model's accuracy is illustrated with several simulations. To use this mathematical description in a digital piano, we are now working on a reduced model. The first result of this reduction, which also can be used for real-time applications, is shown.

Background

Bartolomeo Cristofori is credited as the first to apply a hammeraction to a harpsichord, in 1709. Gradual changes to this mechanism have led to the setup shown in Figure 1. There are many mechanisms in use today, each covered by its own patent.

Electric pianos are more-or-less accepted nowadays as alternatives for traditional pianos. Basic research in piano string behavior (Weinreich 1979) and sound radiation (Suzuki 1986) has led to greater insight into the sound quality of acoustic pianos, which has proven useful in electric pianos. As the sound quality of such digital pianos improves, attention will also turn to the electronics used for scanning. After all, it is through the keys that pianists must express themselves.

There are always differences between grand pianos, and a pianist must adjust his or her performance to the available concert grand. The critical point here is how pianists can get accustomed to a new instrument. They are able to adjust their performance because information flows from the instrument to the performer, via sound, and also through a bi-directional information flow from the haptic senses (tactile, kinesthetic, force, etc.). A more detailed study of the correspondence between the touch response and the sound of an acoustic instrument can be found in Gillespie (1992).

The traditional piano keyboard, which has existed in essentially the same form for the last several centuries, is not yet found in electric pianos. Systems have been developed that measure the time used to depress the electric piano keys, and transform this information into note-velocity information. Figure 2 shows a particular construction scheme, and the time evolution of a "note on-off" event. For this application, application-specific integrated circuits (ASICs) like the E510 in Figure 2 can be used. These integrated circuits also deal with the de-bounce problem, but the scanning technique used in actual digital pianos is not suitable for detecting a wide range of key dynamics. Many companies are also doing research on better key systems, particularly on improving the mechanics. FATAR, an Italian keyboard manufacturer, uses a small hammer action to obtain behavior that is similar to the grand piano's mechanical action. This is actually a passive compensation for the real action. An active, controlled compensation that can replace the mechanics of the action is part of the systems developed at the ACROE Center in Grenoble, France (Cadoz 1990), the Center for Computer Research in Music and Acoustics (CCRMA) at Stanford University (Gillespie 1992), and by independent inventors (Baker 1988). This active compensation is an important contribution to a better key system; however, a simple reduced hammer action or a controlled actuator doesn't resolve the problem completely, because the hammer can move freely and its movement must be simulated.

Computer Music Journal, 19:2, pp. 15–22, Summer 1995

Figure 1. The grand piano key action. In this figure the piano key mechanism is shown in its original rest position. When the key (a) is depressed, it lifts the damper (h) and pushes the whippen (b) upward, rotating it around its pivot point. This moves the jack (c) upward, which pushes against the roller, thus swinging the hammer (f) toward the string (i). The back end of the jack then touches the jack regulator (d), which makes the top end of the jack slip away from under the roller. When the hammer bounces back from the string, it falls on the repetition lever (e), which is still raised by the depressed key, while the back-check (g) prevents the hammer from bouncing up against the string again. The hammer is then in a position much closer to the string than its original rest position. When the key is released slightly, just enough to release the jack from the jack regulator (d), the jack is pulled back under the roller by spring action and the key can be depressed again, throwing the hammer once more toward the string. Because the hammer remains relatively close to the string and the key must be released only slightly, notes can be repeated very rapidly. When the key is released entirely, the whippen, jack, and repetition lever are lowered, allowing the hammer to fall back to its original rest position. A more detailed study can be found in Askenfelt and Jansson (1990). To study the behavior of this action we used position and force sensors; the FUGA CCD camera, which was used in this setup, is part of a family of imaging systems on chip. The FUGA is intended to be interfaced with the outside world without additional external circuitry or post-processing. Image operations are performed in parallel, on-chip. The FUGA can generate 1 million pictures per sec; the output is the number (7-bit) of covered pixels. The force transducer consists of a Wheatstone bridge formed by four strain gages. The signal from that bridge is amplified and filtered. The maximum amplitude is 1 V and the sampling frequency for the 12-bit analog-to-digital converters is 2 kHz.

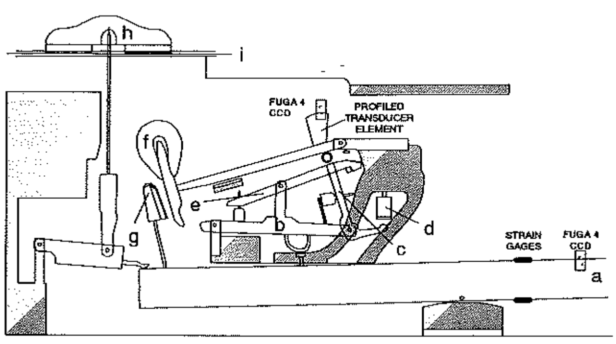
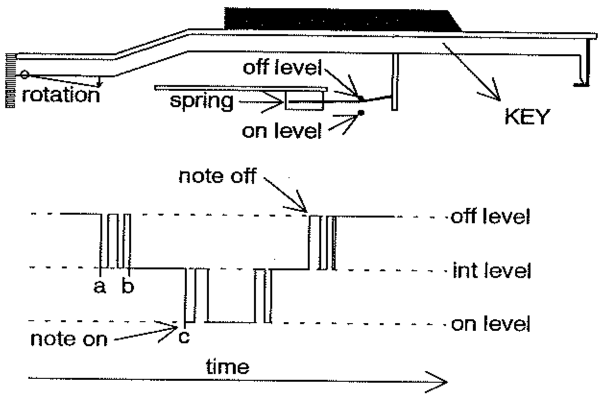


Figure 2. Key velocity measurement system in an electric piano. When a key is depressed at time a*, the spring is moved from the "off" level toward the "on" level. When the spring-off contact is broken (time* b*), a timer is activated. It is stopped as soon as there is a spring-on contact (time* c*). This information is converted into MIDI-compatible velocity information.*



The software model presented here reveals the particular characteristics of a classic piano action. It shows that the behavior of the key and hammer can be simulated in real time, and that this model can be used in an electric piano to improve the key sensitivity. This model can also be used to design an actuator controller.

The Polonaise Experiment

To motivate this research from a pianist's viewpoint, consider highlighting a melody with accents or loudness. This requires good control over all the fingers and a well-tuned action. Another example is changing rapidly from one octave to another (e.g., the finale of George Gershwin's *Rhapsody in Blue*); each time one moves between octaves, there is a new impact of the fingers on the keys. It is extremely difficult to control the sound level when playing this on a digital piano.

To investigate these problems, we must look at the input (the pianist's force) and the output (the hammer displacement). We designed the setup shown in Figure 1 for this purpose. We measure the key and hammer displacement using two optical FUGA4 sensors (Dierckx 1992). Force is measured with a Wheatstone bridge formed by four strain gauges (two on the upper side of the key, and two on the lower side). The strain values have the same

Figure 3. Excerpt from Fryderyk Chopin's Polonaise *op. 40 no 1.*

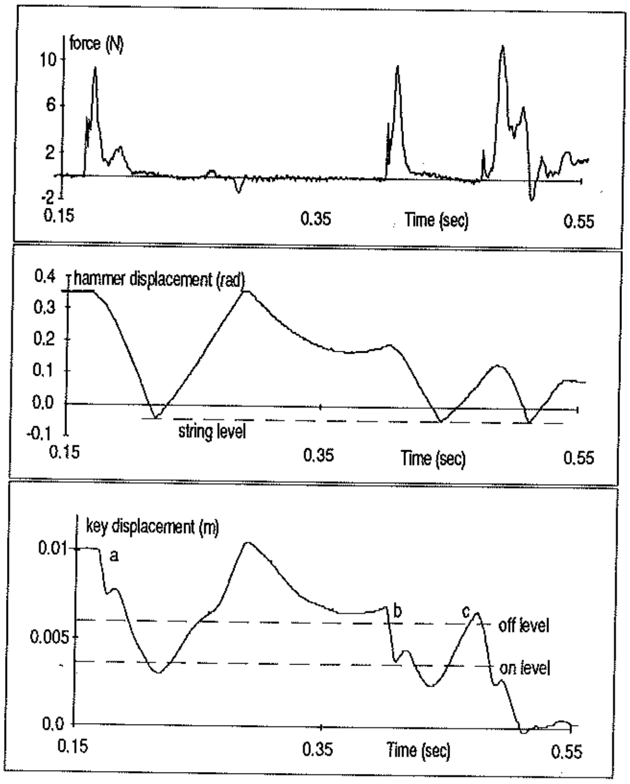


Figure 4. Scan results of the first three chords of the music of Figure 3. The two dashed lines in the bottom frame represent the "off" and "on" levels of the actual scan system. The key for the second "note-on" event (b) was already close to the "off" level; it was depressed slightly. The key (c) for the third "note-on" event also started from a level close to the "off" level. Imagine what would happen if the "off" level were moved somewhat higher. When the key is depressed the second time (b) the delay between the off-on events is clear. This is because the key goes down faster than the finger, due to impact. Next, the key goes up until it meets the finger again, and finally, the key is pushed down. In this case the actual scan system computed an off-on time that was not relevant for that event.

magnitude but opposite signs. An operational amplifier insures a maximum 1-V amplitude bridge signal. For sampling, we use a 12-bit digital-to-analog converter. A profiled transducer element was designed and used, because the hammer rotates around its pivot point and the sensor only detects a linear displacement. A multiple input-output board links this equipment to the computer. Although the FUGA sensors are able to generate 1 million pictures per second, a sampling frequency of around 2 kHz is suitable to investigate the behavior of the key and the hammer.

A typical example shows what the setup's measurements generally look like. When a pianist plays the first three chords of Fryderyk Chopin's *Polonaise* op. 40 no. 1 (Figure 3) on a digital piano, he or she will find it hard to control the sound level of the second and third chord. Some scan results are shown in Figure 4. This figure also illustrates the main deficiency of the actual scan system—if an active mechanical compensation (Cadoz 1990) or a passive one (FATAR's solution) is used in such a way that the key behavior is similar to the real action, and the actual scan technique of electric pianos is used (measurement from two levels), then the hammer velocity computation will be wrong.

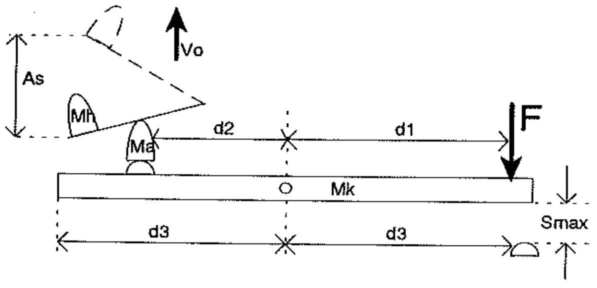
Modeling

The aim of this section is to explain how an improved scan technique can be developed based on a detailed model of the actual system.

A Simple Model

It is not necessary to use complicated situations (as above) to convey the shortcomings of most scan systems. Using a simple model of the action (illustrated in Figure 5), it can be shown (Dijksterhuis 1965) that if a force F is applied to the key, the hammer will have a release velocity v_0, which is the velocity of the hammer after the key has be en pressed down. According to Dijksterhuis the total equivalent mass

Figure 5. Dijksterhuis's action model. m_a *represents the whippen, lever, and jack mass. These masses are reduced to a fixed-point mass. The action springs and dampers are neglected.* m_h *is the hammer mass,* m_k *is the key mass. The key is pushed down over* S_{max} *meters.* F *is the force from the pianist's finger.*



of the key, $m_e = m_{key} + m_{hammer} + m_{action}$, experienced at its excitation point (where the force is applied to the key), is about 0.308 kg. This particular key needs a downward force F_s of 0.45 N at the point of excitation to overcome the spring and friction forces. When a key is excited by a constant force F larger than F_s the hammer velocity, v_0, will be

$$v_0 = v_{hmax} = A_s \sqrt{\frac{2\left(F - F_s\right)S_{max}}{m_e}}$$

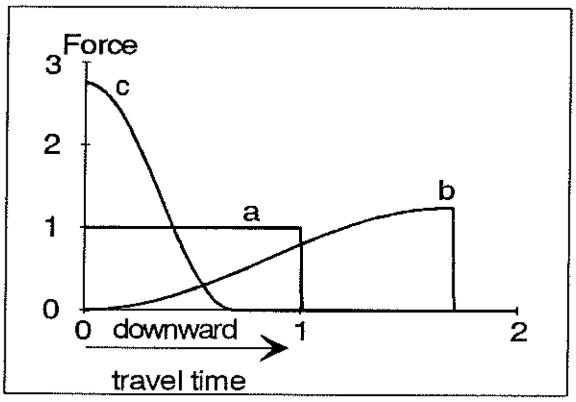
The parameters are shown in Figure 5.

Since pianists almost never play with constant force, it is interesting to consider a situation in which forces vary with time. Choosing (a) ,(b) and (c) (see Figure 6), one concludes that the touch of case (c) will produce a temporally advanced note compared with cases (a) and (b), because the downward travel times (the x-axis) are different. This way, a pianist can emphasize a note without playing it with greater intensity. The key scan system used in actual digital pianos cannot deal with such problems, because they only get the key velocity between two pre-set levels. Therefore more complex systems must be used.

Complex Models

To get an idea of "memory complexity," count the number of "integrators" in a model description. We need explain only a few of these principles to make the conclusions understandable. Consider a key system that only has the ability to detect a "key-on event"—the only two states for such a system are "key-on" or "key-off." To describe the state, a single bit of memory is required for each key. If a key is depressed, then that memory is set to 1; if it is released, the memory is set to 0. In a velocity-sensitive system, the amount of memory must increase. When a key is depressed, the time that passes between the "off" and "on" levels is used to predict the key's velocity. Now one needs at least 1 bit of memory for key-off or key-on states, and several additional bits to measure the delay between the off and on levels. Since MIDI uses 7 bits for key velocity, this seems like a reasonable minimum amount of memory. Note, however, that the situation is somewhat more complicated because of debounce problems. As we explained above, such a system cannot predict the hammer movement with a high degree of accuracy. Referring to Dijksterhuis (1965), we must track the key's movement. To do this, we need a memory for the key position and for the key velocity. If we were to make a functional block diagram we would find two integrators, but for an accurate prediction we need even more. Therefore, we constructed a complex model, which was built hierarchically.

Figure 6. Magnitude of three candidate forces as functions of time. Forces a, b *and* c *are chosen in such a way that tones of exactly the same intensity are produced.*



The Modeling Technique

Using differential equations and state-space techniques to model a system (Wiberg 1971) are beyond the scope of this article. Here we will explain the key ideas of our modeling technique.

Several techniques are described in the literature

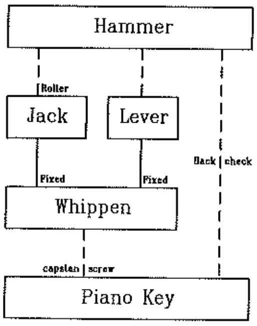


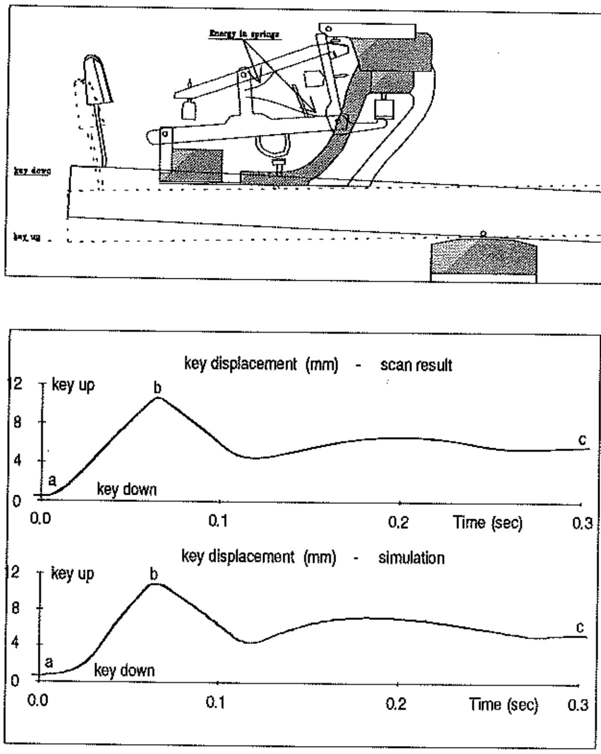
Figure 7. Structural representation of our model. Solid lines represent fixed connections; dashed lines are conditional connections.

Table 1. Complexity of the various components of the program model

Action Element	*No. of Parameters Equations*	*No. of Differential Equations*	*No. of Constraint Equations*
Whippen	14	10 × 2nd order	8 + 1 implicit
Jack	10	7 × 2nd order	5 + 1 implicit
Key	14	8 × 2nd order	6 + 1 implicit
Lever	10	6 × 2nd order	4 + 1 implicit
Hammer	10	6 × 2nd order	4 + 1 implicit

for modeling physical dynamics systems. In general, engineers are most familiar with block diagrams and differential equations. Major drawbacks of these methods are the lack of structure during the model generation phase, the absence of the physical system's structure in the generated model (Karnopp 1989), and the causality problems that can arise when subsystems must be coupled. When dealing with small dynamic systems, these negative points introduce very few problems; however, for a large, dynamic system such as a piano key action, the classical methods are not always error-free. Our model was hierarchically constructed using bond graphs to maintain clarity during the construction process (Karnopp and Rosenberg 1975). A structural representation is shown in Figure 7. The complexity of our detailed model was analyzed by translating it into a block diagram, and counting the differential equations.

Figure 8. Key release is simulated without the hammer. The top figure shows the system without the hammer. First the key is pushed down completely (solid lines). Because of spring compression, energy is stored in the action. After a while the key is released, at which time both the scan and simulation are started (time a). The action pushes the key upward (b) but it takes approximately 300 msec before the position of the key becomes stable (c). The key stays in the middle between the 0 (key down) and 0.01 (key up) level because the action has no hammer.



They are not included here, but can be found in Van den Berghe (1992). We used the dynamic-system simulation program, DYNAST (Dynast 1992). In this application, macros were created for each action element (key, whippen, jack, lever, and hammer). Such a macro consists of: (1) parameters: dimensions, inertia, and mass; (2) differential equations; and (3) constraint equations. The action components are modeled as stiff systems; because of this stiffness, there are additional equations. The total number of equations for these five macros are shown in Table 1.

The whippen, key, and hammer are fixed to the frame, and the jack and lever are fixed to the whippen (implicit equations). For each of these macros, the interface with the outside world is its node. At the top of the hierarchy the same nodes are repeated, to make reference to these macros. In

Figure 9. Pushing down a key without producing a sound. If a small force is applied to the key (time a*), the key reaches the bottom frame after approximately 20 msec (*b*). As one can see, the force applied to the key is too small to throw the hammer against the string, so there is no hammer-string contact. When the hammer falls, it falls onto the repetition lever. In the model, which was used to simulate this event, the back-check effect is not included, and therefore the hammer is not held in a fixed position when the key is depressed (*c*). When the back end of the jack touched the jack regulator, there is a discontinuity in the key displacement (*d*). This happened when the jack slipped away from under the roller.*

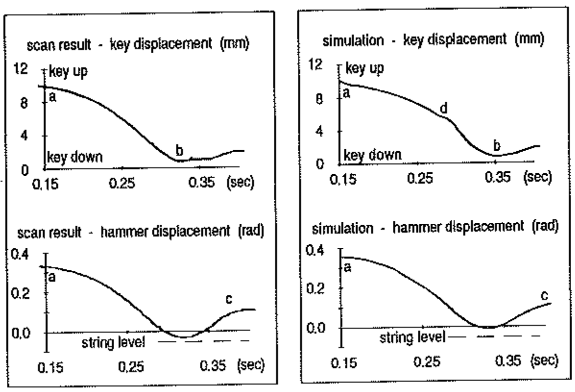


Figure 10. Impact from a finger on a key. The most spectacular event occurs when there is impact from a finger on the keys. First the key goes down faster than the finger (time a*) because of the impact, next the key goes up until it meets the finger again (*b*) and finally the key is pushed down completely (*c*). This is the most important problem, because it is the main deficiency of the actual MIDI velocity scan system. For a good key system, the model must predict when the hammer-string contact takes place (*e*). Even when an active controlled compensation for the action is used, a model to predict the hammer movement is necessary because the hammer can move freely, depending upon the force that is applied to the key. One can see that for this simulation the back-check effect is part of the model description (*f*). The phenomena when the jack slips away from under the roller can also be noticed (*d*).*

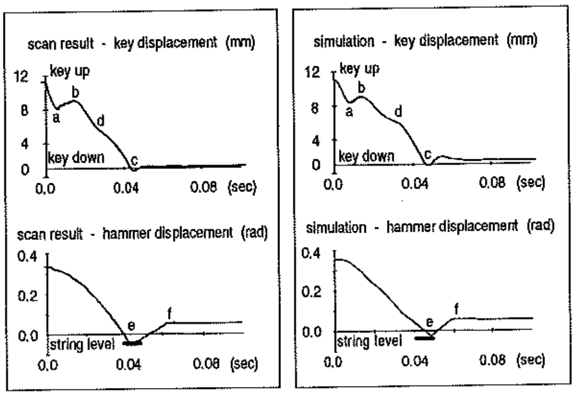


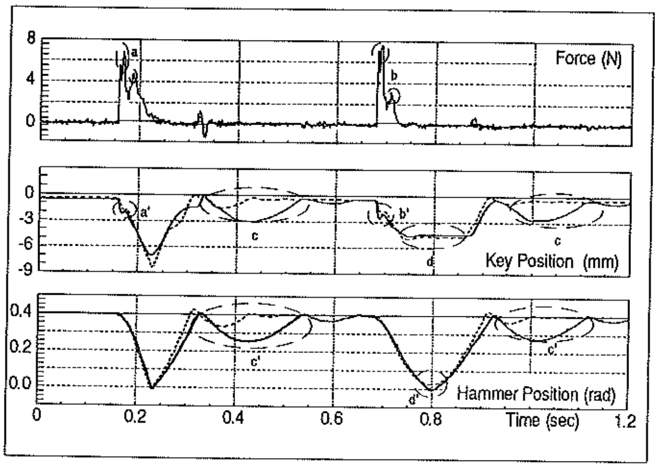
Figure 7 the top level of the hierarchy is represented. Some of the connections between these macros are fixed, others are conditional. At the top level, non-linear models are defined for the felt between key and whippen; jack, lever, and whippen; hammer and jack; hammer and frame; key and frame; and lever and frame. The action springs are also modeled as non-linear parts. This top model has 22 parameter descriptions and 14 extra equations to model felt and springs. The model's accuracy is illustrated in Figures 8, 9, and 10. More details about this model can be found in Van den Berghe (1992).

Continued Research

As one can see from Figures 8, 9, and 10, we were able to closely model the action characteristics. Most of the simulations were accurate. To execute this model in real time would require a super-computer. The simulation software DYNAST has a variable time-step integration algorithm. Because ours is a stiff system and the action topology is changing during simulation (e.g., depending on the position of the action, the hammer looses contact with the jack), a simulation of 1 sec in real life takes about 4 hours on a 486DX33 IBM PC-compatible computer. This is one reason why we are working on a reduced-parameter model.

In carefully analyzing the situation, we noted the influence of the parameters, and eliminated those that were not relevant. In the complex model, which was developed first, every parameter and equation had a physical meaning. Because of the hard non-linear characteristics of the system, we could not use classic linear model-reduction techniques. Therefore we represented the system with a smaller number of non-linear equations, and trained the parameters with experiments. At the same time, we have transformed the model into a discrete-time model. A first result is shown in Figure 11. As there was a need for a more powerful simulation package, we used ISI's System Build Environment (System Build 1993). Running on a Digital Equipment Corp. DEC 5000/33 workstation, the simulation of 1 sec in real life took

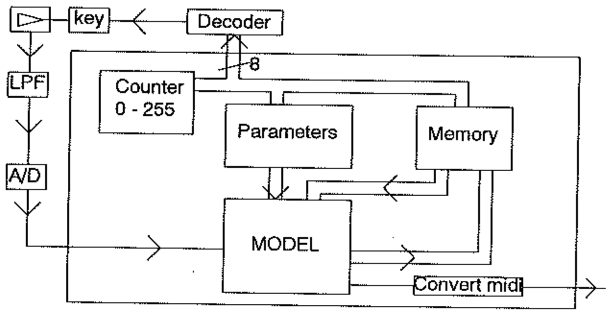
*Figure 11. The reduced model simulation. Two applied forces are in the top frame (*a *and* b*). The measurement (solid line) and simulation (dashed line) of the key position is shown in the middle frame. The bottom frame shows the hammer position. Each force has two impact phenomena—due to the impact, the key goes down faster than the finger (*a' *and* b'*). Although there is only a small difference in applied force, the key movement is quite different. In the first case (*a *and* a'*) the key reaches the bottom; but in the second case the key stays somewhere in between (*d*). One can verify that the simulation of the key-down and hammer-up events is accurate. This is especially true for* d *and* d'*—the key is hardly depressed, but the virtual hammer (part of the model) reaches the string at almost exactly the same time as the real hammer. The model is part of a virtual system: for each key the virtual hammer position is simulated. Because we are not limited to real physical constraints, we can specify some parameters of the model to give the virtual system a more desired behavior—the key release of the virtual system (dashed lines) needs less time, and the stable position is reached faster (*c *and* c'*). The same is true for the hammer—the sooner this stable position is reached, the more control we have over the next key-on event.*



about 15 sec with the reduced model. Our next step is to represent the system with a neural network. Using this representation we can apply more powerful training techniques, and we end up with a fully parallel system, which will reduce the simulation time.

This reduced model can be implemented using an ASIC. A schematic representation of an ASIC using such a reduced model is shown in Figure 12. If we were to make a digital piano using such key models, we would be able to investigate the behavior of pianists when some parameters of the model are changed. By changing only the parameters, pianists will presumably think that the mechanism itself is changed (e.g., a different hammer). Such effects can be investigated with very little effort, since there are only a few parameters to be changed. A first reduced model, and some implementation aspects, can be found in Van den Berghe (1993). The software simulation device we have built may prove extremely useful for designing a better piano key action for electric pianos such as the Yamaha and Roland, and especially for companies whose pianos use a FATAR key action.

Figure 12. ASIC structure. First a counter specifies the key number. The action characteristics might be different from one key to another, and each key model uses different parameters. When a key is selected, the previous state (from the memory) together with the key's new position are used to compute the next state. Depending on that state, an output can be generated (output = "key on" [+ velocity] or "key off"). A decoder, outside the ASIC, translates the counter signal into a single key identification. From that key, the analog signals (key position levels) are converted into a digital signal, (the "LPF" is a low-pass filter, which is fed into the analog-digital [A/D] convertor); this is then the input for the ASIC. A de-bounce algorithm is used to avoid needless state computations.



Conclusions

The actual scan system used in most synthesizer keyboards today is not capable of predicting the hammer velocity with any accuracy. It has been shown that the scan system must track the key displacement. Using a detailed mathematical model, we can improve the sensitivity of electric pianos. To do this, the model must be reduced, the parameters must be trained from experiments, and the algorithm must be implemented in an ASIC.

References

Askenfelt, A., and E. V. Jansson. 1990. "From Touch to String Vibrations. I: Timing in the Grand Piano Action." *Journal of the Acoustical Society of America* 88(1):52–63.

Baker, R. 1988. Active Touch Keyboard. United States Patent No. 4,899,631.

Cadoz, C. et al. 1990. "A Modular Feedback Keyboard Design." *Computer Music Journal* 14(2):47–51.

Dierckx, B. 1992. *The Fuga Series of Intelligent Imagers.* Available from IMEC v.z.w, Kapeldreef 75, B-3001 Leuven, Belgium.

Dijksterhuis, P. 1965. "De Piano." *Nederlands Akoestisch Genootschap* 7:50–65.

Dynast 1992. *Reference Manual.* Available from DYN, Nad Lesikem 27, CS-16000 Prague 6, Czech Republic.

Gillespie, B. 1992. "Dynamical Modeling of the Grand Piano Action." *Proceedings of the 1992 International Computer Music Conference*. San Francisco: International Computer Music Association. pp. 77–80.

Karnopp, D. C. 1989. "Structure in Dynamic System Models. Why a Bond Graph is More Informative than its Equations." *IMACS Transactions on Scientific Computation* 88(3):15–18.

Karnopp, D. C., and R. C. Rosenberg. 1975. *System Dynamics : a Unified Approach.* New York: Wiley.

Suzuki, H. 1986. "Vibration and Sound Radiation of a Piano Soundboard." *Journal of the Acoustical Society of America* 80:1573–1582.

System Build. 1993. *Reference Manual.* Integrated Systems Inc., 2500 Mission College Blvd., Santa Clara, California, USA.

Van den Berghe, G. 1992. "Modeling a Concert Grand Action." Master of Science Thesis. ESAT, Katholieke Universiteit Leuven, Kard. Mercierlaan 94, B-3001 Leuven, Belgium.

Van den Berghe, G. 1993. "Piano Key Action Models for Electric Pianos." *The IEEE 1994 Student Paper Book.* New York: IEEE Press.

Weinreich, G. 1979. "The Coupled Motions of Piano Strings." *Scientific American* 240(1):94–102.

Wiberg, T. 1971. "Theory and Problems of State-Space and Linear Systems." New York: McGraw-Hill.