ViP: Controlling the Sound of a Piano with Wrist-Worn Inertial Sensors

Aristotelis Hadjakos
Zentrum für Musik- und Filminformatik
HfM Detmold / HS OWL
Hornsche Str. 44
32756 Detmold, Germany
hadjakos@hfm-detmold.de

Simon Waloschek
Zentrum für Musik- und Filminformatik
HfM Detmold / HS OWL
Hornsche Str. 44
32756 Detmold, Germany
waloschek@hfm-detmold.de

1. INTERACTION

The ViP (**Vi**brato **P**iano) lets the user influence the pitch of played notes by rocking the hand left and right along the keyboard while keeping the notes to be modified depressed. For this purpose, the user wears an inertial sensor on each wrist. While we use a custom-built sensor solution, the principle described in this paper could also be realized using smart watches, making the instrument accessible to a large user base. We process the sensor signals in a way that we add the vibrato capability without breaking traditional playing patterns such as chords, passages, jumps, or tremolos by avoiding unwanted vibrato artifacts. Each hand's rocking movements control the vibrato of the notes grasped by that hand. For that purpose we employ an algorithm that assigns notes to hands based on two Kalman filters operating on MIDI data.

2. IMPLEMENTATION

Figure 1 provides an overview of the implementation. We use inertial sensing to capture the player's wrist movements (see Section 2.1) and process the acceleration signal to provide a natural playing experience (see Section 2.2). Furthermore, it is necessary to know which hand plays which notes so that the vibrato movements of a hand controls only the notes it is currently grasping. We use a MIDI-based method to assign notes to hands (see Section 2.3). The synthesis is then performed separately for each hand. We currently use a subtractive synthesis method based on a bank of pink noise generators and resonating filters that generate the sound based on individual envelopes for each generator-filter pair.

2.1 Inertial Sensing

For inertial sensing we use a breakout board with the MPU-9150 sensor chip made by InvenSense [3]. The chip has a 3D accelerometer, a 3D gyroscope, and a 3D magnetometer. It provides the measurements via I^2C. We use our SPINE platform [2] (see Figure 2), a hybrid between programmable physical computing platform and easy-to-use TUI toolkit, to forward the values from the MPU-9150 to the computer. The computer then processes the signal and performs the sound synthesis.

Keyboard Salon @ NIME'14, June 30 2014, Goldsmiths, University of London, UK.

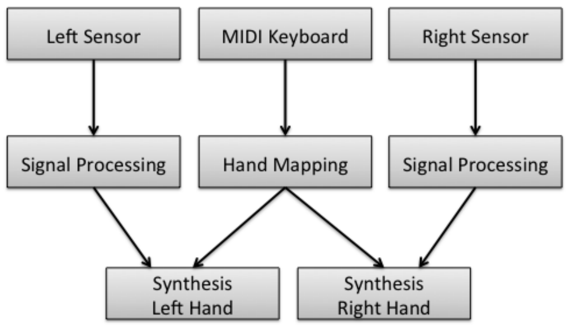

Figure 1: ViP implementation overview

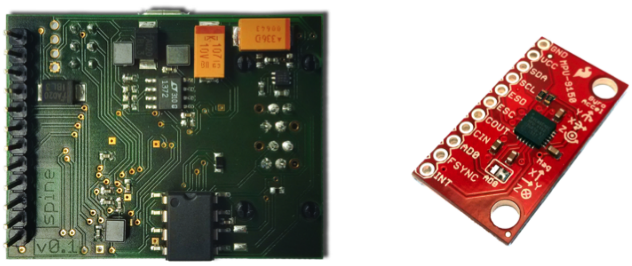

Figure 2: The SPINE (left) is used to interface with the MPU-9150 inertial sensor breakout (right)

2.2 Signal Processing

The goal of our signal processing pipeline was to add the ability to control a vibrato while still making it as easy as possible for the players to play without vibrato when they choose to. Among other things it should be possible to play passages, chords, jumps and tremolos (see Figure 3) without undesired vibrato artifacts.

Filtering: To control the vibrato, the user presses down the keys and rocks the hand left and right while still keeping the notes depressed. To capture that movement we use the y-axis of the accelerometer signal, which is pointing to

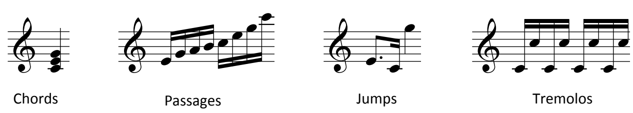

Figure 3: Musical patterns that can be played without vibrato artifacts.

the side of the user's forearm along the keyboard. This signal oscillates when performing the rocking movement. Due to the pull of gravity, the accelerometer signal can include a constant offset if the user does not hold his or her arm completely level. Since the signal shall be used to detune the pitch of a synthesis patch, we use a high pass filter with a cutoff frequency of 0.5 Hz to remove the constant component in the signal. To remove discontinuities in the sensor signal we also pass it through a low pass filter with a cutoff frequency of 50 Hz, which is a high enough frequency value in order to avoid filtering out the user's vibrato movement. The resulting signal is clipped to the range of ± 1 g and is used to detune the synthesis patch by the amount of ± 1 semitones.

Sensitivity ramping: When a note is just being played (i. e., when the "note on" message is received) the note is played without detune. The influence of the detune is then ramped up from 0 to 100% over the course of 200 ms. When the note is released the influence of the detune is immediately set to zero while the note may still be sounding for a short period of time in the release phase.

Discussion: The signal processing ensures that traditional patterns can be played without introducing (too much) unwanted vibrato artifacts. Basically a normal execution of a traditional playing pattern should result in as little as possible vibrato:

Chords: When a single note or a chord is played, a reaction force is generated when the key hits the keybed. This reaction force is transmitted through the finger and the hand up to the wrist. The effects of that reaction force can be sensed by a wrist-worn accelerometer. Most of the resulting acceleration is present on the vertical axis but a smaller part can also be sensed on the y-axis that we use. Furthermore, the arm may participate and support the finger with a downward movement. Again a part of the downward acceleration is sensed along the y-axis.

Passages, jumps, and tremolos: Passages, i. e., successions of rather quick scales and arpeggios, typically show acceleration peaks in the y-axis signal that we use due to horizontal hand re-positioning. These acceleration peaks are also very common in jumps and tremolos.

All these situations are covered by the sensitivity ramping keeping the level of unwanted vibrato low.

2.3 Hand Mapping

The vibrato movements of each hand should only modify the pitch of the notes kept by that hand. In order to do that it is necessary to know which notes were played by which hand. For this purpose we use our MIDI-based method described in [1]:

The algorithm assumes that the hands do not cross over each other. It keeps track which notes are currently held down. In the stream of incoming notes, some notes can then be uniquely assigned to the left or right hand if they are spaced so far apart that they cannot be grasped by a single hand. Such "unique notes" are assigned to the left respectively right hand. Furthermore, the algorithm has an internal model of the hand positions. The hand positions are modeled as two Gaussian probability distributions, described by the center of the hand (a fractional MIDI note value) and the variance of the distribution. When a note is played, which is not recognized as a "unique note" it is assigned to the closest hand based on the current hand position predictions. One Kalman filter per hand is used to update the estimates of the hand positions: When a note is assigned to one hand, it modifies the predicted center position and also decreases the variance of the probability distribution. An immediately following MIDI note assigned to the same hand would then have a lesser effect on the center position so that the predicted hand position would tend to converge on the actual position of the hand. The variance of the probability distribution increases with time in a constant rate, allowing for hand position changes and jumps.

An evaluation of the algorithm shows very good accuracy rates of 94.3 to 99.4% in classical piano pieces (Bach Sinfonias 1-5, BWV 787-791), which are markedly better than the naïve method to split the keyboard into two areas at the center C key [1]. The occasional wrong detections typically occur in inner voices where the two hands are close to each other. The vibrato effects are most noticeable in the uppermost and lowermost notes, making wrong detections in the inner voices less problematic.

3. USING THE VIP

Demonstration: The participants of the workshop will be able to use and play the ViP.
Setup: The inertial sensors have a size of about 1.9 x 2.8 cm and are worn on the left and/or right wrist. The sensors are insulated with a shrink down plastic tube and are connected to the SPINE platform with a cable. A sweatband is used to clamp the sensor to the player's wrist.
Technical requirements: A MIDI-enabled keyboard with 88 keys is needed.

4. REFERENCES

[1] A. Hadjakos and F. Lefebvre-Albaret. Three methods for pianist hand assignment. In *SMC*, 2009.
[2] A. Hadjakos and S. Waloschek. SPINE: a TUI toolkit and physical computing hybrid. In *NIME*, 2014.
[3] InvenSense. MPU-9150 product specification, revision 4.0, 2012.