



**Weitere Interessante technische Informationen zum Thema
Audiorecording finden Sie auf der Seite von [RME](#)**

**Low Latency Hintergrund: Von Puffern und Latenz Jitter
oder: Was Sie schon immer über Latenz wissen wollten (oder auch nicht...)**

Einleitung

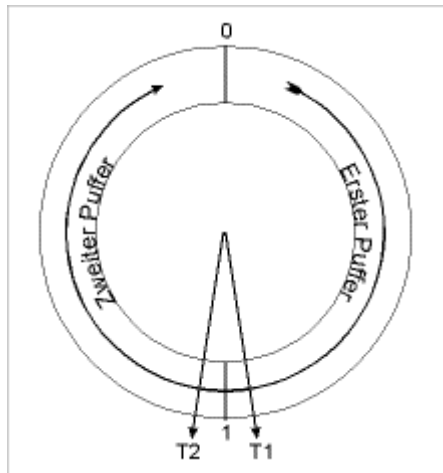
Latenz Jitter bezeichnet Schwankungen in der Latenz, also gewissermassen der Reaktionszeit des Systems. Der Latenz Jitter ist ein bisher praktisch vollständig ignoriertes Thema, welches gleichwohl eine der grössten Herausforderungen der Softwareindustrie für die nächste Zeit sein wird. Zuerst die gute Nachricht: Bei gleichzeitigem Aufnahme- und Wiedergabebetrieb ist die Latenz konstant. Das Phänomen der variierenden Latenz ergibt sich beim derzeit so beliebten Einsatz von Software-Synthesizern und Samplern, also einer von ausserhalb des Systems gestarteten Klangerzeugung.

Theorie...

Um die Entstehung des Effektes zu verstehen muss man wissen, dass die Kommunikation zwischen Audiohardware und Software praktisch ausnahmslos nicht als konstanter Datenstrom, sondern in einzelnen Abschnitten oder Blöcken, nämlich den bereits früher erwähnten Puffern, stattfindet. Daten werden vom Anwendungsprogramm immer im Voraus in einen Puffer kopiert, und dieser von der Audiohardware später abgespielt.

Schritt für Schritt: Das Diagramm unten zeigt exemplarisch einen (wie in ASIO zu findenden) Doppelpuffer. Der geschlossene Kreis verdeutlicht den Ablauf, da in einer ständigen Wiederholung nach dem zweiten Puffer wieder der erste folgt. Jeder Puffer fasst hier 256 Samples, entsprechend einer Latenz von 6 ms. Wie diese 6 ms entstehen ist einfach nachzuvollziehen. Beim Start der Wiedergabe befindet sich der Positionszeiger an der mit 0 bezeichneten Stelle. Da beim Start der Wiedergabe der erste Puffer bereits abgespielt wird kopiert die Software alle Daten zunächst in den zweiten Puffer. Stellen Sie sich (in Zeitlupe) vor, wie sich der Zeiger im Uhrzeigersinn fortbewegt. Bei einer Samplefrequenz von 44,1 kHz werden die 256 Samples in nur 5,8 ms abgearbeitet. Sobald der Zeiger die Grenze zwischen den Puffern erreicht hat (Position 1) wird ein Interrupt ausgelöst, und die Software kopiert die nächsten Daten in den (nunmehr abgearbeiteten) ersten Puffer. Der zweite Puffer wird nun abgespielt, am Ende (Position 0) wieder ein Interrupt ausgelöst, Daten in den zweiten Puffer kopiert usw.

Damit ist klar: Alle von der Software erzeugten Daten werden frühestens nach Durchlauf eines kompletten Puffers ausgegeben - womit sich die besagte Latenzzeit ergibt. Und schön konstant ist sie auch.



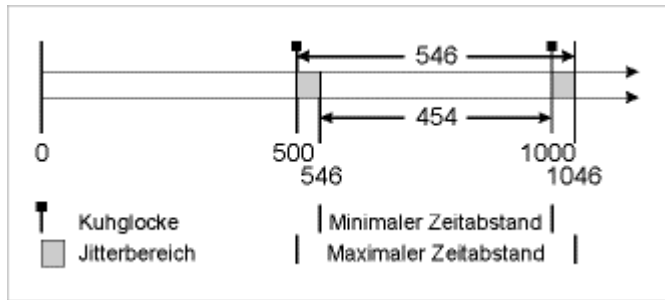
Interessant (und ein wenig komplizierter) wird es, wenn nun statt einer Wiedergabe von Festplatte (gleichmäßiges Streaming) an einem MIDI-Keyboard zu unvorhersehbaren Zeitpunkten eine Taste gedrückt wird. Beispiel 1: Der Tastendruck erfolgt bei der Zeigerposition T1, kurz vor einem Pufferwechsel. Die Software berechnet sofort einen Klang, der Zeiger überquert kurz darauf die Puffergrenze und löst eine Übertragung der berechneten Daten in den ersten Puffer aus. Nach Überquerung des zweiten Puffers wird der erzeugte Klang abgespielt, mit einer Verzögerung von einem Puffer plus der minimalen Zeitspanne zwischen T1 und dem Pufferwechsel 1. Die dabei entstehende Zeitveränderung ist sehr gering und bestenfalls ein Stirnrunzeln wert.

Das ändert sich mit Beispiel 2: Der Tastendruck erfolgt bei der Zeigerposition T2, kurz nach einem Pufferwechsel. Die Software berechnet sofort einen Klang, der Zeiger braucht jedoch nun allein fast 6 ms um erst einmal die nächste Puffergrenze zu erreichen und das Kopieren der Daten in den zweiten Puffer auszulösen - der aber erst nach weiteren 6 ms abgespielt wird (erst einmal ist noch der erste Puffer dran). Damit wird der 'in Echtzeit' erzeugte Klang von der Audiohardware erst nach knapp 12 ms wiedergegeben! Als Faustformel gilt daher derzeit unter ASIO: Bei Einsatz von Software-Synths und -Samplern schwankt die reale Verzögerung bis zur Ausgabe zwischen einer und zwei Puffergrößen. Die Höhe der Schwankungsbreite ändert sich zum einen mit der Puffergröße selbst, zum anderen mit dem verwendeten Verfahren. Das von Gigasampler her bekannte GSIF-Interface beispielsweise arbeitet mit 3 Puffern à 128 Samples, von denen zwei fester Anteil sind. Damit ergeben sich effektive 6-9 ms Latenz.

...und Praxis

Sicher fragen Sie sich ob dies alles nur theoretischer Firlefanz, oder auch in der Praxis nachzuvollziehen ist. Dazu ein simples, aber überzeugendes Beispiel. Stellen Sie die verwendete ASIO-Audiokarte auf höchste Latenz (beispielsweise 46 ms). Spielen Sie nun mittels MIDI-Keyboard Metronom, also einen perkussiven Klang (z.B. Kuhglocke) mit gleichmäßigen Anschlägen von circa 120 bpm. Was jetzt aus dem Lautsprecher stolpert hat mit dem Keyboardspiel nichts mehr gemein. Eigentlich sollten die Kuhglocken genauso aus dem Lautsprecher kommen wie sie gespielt werden, nur halt mit der erwarteten Verzögerung. Wegen des Latenz Jitters schwanken sie jedoch um 46 ms, sind also mal 454 ms und mal bis zu 546 ms auseinander. Das entspricht einer Tempovariation von 110 bis 132 bpm, und ist unüberhörbar.

Jetzt kommt's dicke: Vorhin war noch von einer Schwankungsbreite von einem Puffer (46 ms = plus/minus 23 ms) die Rede, jetzt sind es plötzlich plus/minus 46 ms. Eine Erklärung liefert das Zeitdiagramm, in dem auf der Zeitachse die ursprünglichen Tastenanschläge im Abstand von 0,5 s zu sehen sind. Die Grundverzögerung von 46 ms wurde der Übersicht halber weggelassen.



Das graue Feld zeigt den Zeitraum, in dem das Audiosignal aufgrund des Latenzjitters von der Audiohardware ausgegeben wird, eben nach 500 oder bis zu 546 ms. Im 'Worst Case' wird Kuhglocke 1 nach 546 ms wiedergegeben, die nächste dagegen bei 1 Sekunde. Damit beträgt der Zeitabstand von Kuhglocke 1 und 2 jedoch nur noch 454 ms, die Wiedergabe erscheint quasi beschleunigt.

Doch Latenz Jitter muss nicht sein. Es ist zwar derzeit (leider) Standard, jedoch nicht unabwendbar. Tatsächlich könnte die Software alle auszugebenden Audiodaten auf das Maximum von 2 Puffern verzögern. Die Latenz für Echtzeit-Synthies wäre damit immer doppelt so hoch, dafür aber vollkommen schwankungsfrei. Wie eingangs bereits erwähnt ist dieser Zustand deutlich angenehmer als wackelndes Timing. Nach uns vorliegenden Informationen war die Firma emagic Vorreiter dieser Technik, in Logic 4.2 gab es keinen Latenzjitter. Da für das aktuelle Logic 4.5 jedoch die Audio-Engine komplett erneuert wurde ist der Jitter wieder da. Zweifellos werden in Kürze Updates von emagic und Steinberg (Cubase 5.0 jittert auch) für Abhilfe sorgen.

Mathestunde

Die Berechnung der Latenz kann einem schon so manches Kopfzerbrechen bereiten - und das nur weil einige Programme die Latenz nicht Anwender-gerecht anzeigen. Generell wird die Puffergröße entweder in Bytes oder in Samples angegeben. Letzteres ist das einzig Wahre, denn die Latenz ergibt sich dann durch einfache Division mit der aktuellen Samplefrequenz:

8192 Samples durch 44100 ergibt 186 ms, durch 48000 noch 171 ms usw.

Bei einer Angabe in Bytes muss jedoch zunächst die zugrunde liegende Kanalzahl, die Anzahl der Puffer sowie die Wortlänge berücksichtigt werden. Beispiel WaveLab Stereo:

4 mal 4096 gleich 16384. Pro Kanal die Hälfte, gleich 8192. Bei 16 Bit Auflösung werden 2 Bytes pro Sample benötigt (1 Byte = 8 Bit). Effektiv sind also 4096 Samples im Puffer, ergibt 93 ms bei 44,1 kHz.

Bei 24 Bit sind 3 Bytes notwendig, demzufolge reduziert sich die Pufferwirkung auf 62 ms.

Im sogenannten ungepackten 32 Bit Betrieb, in WaveLab '24 Bit alt.' genannt, werden 4 Bytes pro Sample verbraucht (obwohl in den meisten Fällen trotzdem nur 24 Bit Audodaten übertragen werden), die erzeugte Latenz beträgt damit nur noch 46 ms.

Zu beachten ist, dass die angegebene Latenzzeit im Allgemeinen der effektiven Buffergröße (Puffergröße) entspricht, und zwar von Record ODER Playback. Die Gesamtdurchlaufzeit durch den Rechner ist also immer doppelt so groß (Record plus Playback). Im Gegensatz zu reiner Wiedergabe ergibt sich für das Monitoring eines Eingangssignales also ein doppelt so hoher Wert.

Aufgrund der inzwischen erreichbaren sehr kleinen Latenzen kann oftmals auf Techniken wie RME's Zero Latency Monitoring verzichtet werden. Generell muss bei Harddisk Recording mit vielen Spuren jedoch weiter mit grösseren Puffern gearbeitet werden. ZLM wie auch ADM bleiben damit bis auf weiteres unverzichtbar.

Copyright © Matthias Carstens, 2000. Alle Angaben in diesem Tech Info sind sorgfältig geprüft, dennoch kann eine Garantie auf Korrektheit nicht übernommen werden. Eine Haftung von [RME](#) für unvollständige oder unkorrekte Angaben kann nicht erfolgen. Weitergabe und Vervielfältigung dieses Dokumentes und die Verwertung seines Inhalts sind nur mit schriftlicher Erlaubnis von [RME](#) gestattet.

Warenzeichen Alle Warenzeichen sind Eigentum ihrer jeweiligen Inhaber. RME und DIGI96 sind eingetragene Marken von [RME](#) Intelligent Audio Solutions. ASIO ist ein Warenzeichen der Steinberg Soft- und Hardware GmbH. Microsoft, Windows, Windows 98/ME/2000 sind registrierte oder Warenzeichen der Microsoft Corp.

Die Freigabe für die Platzierung dieses Atrikels auf der Homepage "Musik-Mit-PC" wurde am 19.November 2000 von Herrn Matthias Carstens, [RME](#), erteilt !