

# Het TimeTossing principe

*Een studie naar de inzetbaarheid van het gebruik van signaalvertraging in ritmisch gekwantiseerde tijdeenheden als een muzikaal instrument.*

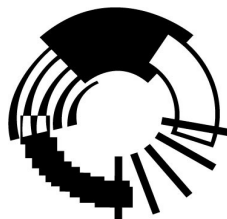
Walther, T.

Hogeschool van de Kunsten Utrecht, faculteit Kunst en Media  
Technologie

Begeleider: Groenewegen, M.

Afdeling EMMA

Utrecht, 10 augustus 2007



**EMMA**  
EUROPEAN MEDIA MASTER OF ARTS

## **Abstract**

This thesis consists of a study on the use of the TimeTossing principle as a musical instrument. This principle is defined as the process of re-arranging a musical audio signal by means of introducing a delay with a variable rhythmically quantized length.

Apart from defining this principle, the design for a hardware prototype incorporating this principle is proposed. This design is based on a number of assumptions such as the user-group, as well on how the user can benefit the most from the user interface with the assumed practical applications in mind.

In an attempt to validate these assumptions, the practical uses of the TimeTossing principle are researched by means of comparing similar musical phenomena. A research in similar existing solutions to the TimeTossing principle aims to provide an indication of the validity of the principle.

To further validate the assumptions made on the user-group and the design, a user-test is carried out with DJ's and Musicians.

# Inhoudsopgave

	<b>Voorwoord</b>	4
<b>1</b>	<b>Inleiding</b>	4
1.1	Toepassingen van geluidssignaalvertraging	6
1.2	Het TimeTossing principe	6
1.3	Het TimeTossing prototype	8
1.3.1	Implementatiecriteria	8
1.3.2	Ontwerpkeuzes	9
1.3.3	Relevantie en vraagstelling	12
1.3.3.1	Formulering van hoofd-onderzoeksvraag	12
1.3.3.2	Formulering van sub-onderzoeksvragen	12
<b>2</b>	<b>Literatuuronderzoek</b>	13
2.1	Methode van onderzoek	13
2.2	Vergelijkbare bestaande oplossingen	13
2.2.1	Resultaten	13
2.2.1.1	Hardware	13
2.2.1.1.1	Samplers	13
2.2.1.1.2	Echo-apparaten	15
2.2.1.2	Software	18
2.2.1.2.1	<i>Beat-Slicers</i>	18
2.2.1.2.2	Supatrigga	19
2.2.1.2.2	Modulaire audio softwarepakketten	19
2.2.3	Conclusie	20
2.3	Vergelijkbare muzikale fenomenen	
2.3.1	Resultaten	
2.3.2	Conclusie	
<b>3</b>	<b>Kwalitatief onderzoek</b>	
3.1	Methode van onderzoek	
3.2	Resultaten	
3.3	Conclusie	
<b>4</b>	<b>Conclusie</b>	
	<b>Literatuurlijst</b>	21

## Voorwoord

Een paar jaar geleden was ik met twee medestudenten op mijn kamer aan het experimenteren met het programma Reason. We waren vooral bezig met de audio delay module van het programma. Deze module stelt de gebruiker in staat om het inkomende audiosignaal te vertragen. De tijdsduur van deze vertraging kon naast muzikale eenheden ook in milliseconden ingesteld worden. Zoals wij al eerder hadden geleerd op onze studie zorgt een kleine vertraging samen met terugkoppeling van het vertraagde signaal terug naar de ingang van de module voor een resonantie, het zogeheten Karplus-Strong effect.

Om dit fenomeen muzikaal te kunnen uitbuiten gebruikten wij meerdere delay modules in deze configuratie en gaven ze hetzelfdeingangssignaal. Echter, de ingestelde vertragingstijd verschilde bij elke module zodat elke module een andere resonantie produceerde. Alle uitgangen van de delay modules werden aangesloten op het virtuele mengpaneel van Reason. De schakelaars die de individuele kanalen aan en uit schakelden werden gekoppeld aan het toetsenbord van de computer, wat een snelle bediening mogelijk maakte.

Doordat één van de modules nog steeds stond ingesteld als ritmische vertragingseenheid, weerklonk elke gespeelde noot één achtste later in de tijd. In een flits besepte ik me dat als ik de andere modules instelde met meervouden van die ene achtste, dat het dan mogelijk zou zijn het ingaande signaal in volgorde te veranderen door de kanaalschakelaars van het mengpaneel te "bespelen".

Een paar maanden later gebruikte ik dit principe in mijn eigen eerste VST plug-in, de Slic0r. Deze plug-in was het eindwerkstuk van een project op de faculteit en werd goed ontvangen. Realiserende dat het principe op elk muzikaal signaal toegepast kon worden, groeide bij mij het idee om dit principe toe te passen in een klein apparaatje wat makkelijk meegenomen en aangesloten zou kunnen worden zonder dat hier een computer bij nodig zou zijn.

Tijdens mijn stage bij stichting STEIM kwam ik meer en meer in aanraking met digitale elektronica. Rene Wassenburg, de hardware/firmware ontwikkelaar van STEIM in die tijd, heeft veel vragen van mij kunnen beantwoorden en heeft mij daardoor laten inzien dat het voor mij haalbaar was om het principe van de Slic0r plug-in te kunnen implementeren in hardware.

Vanaf dat moment was ik vastberaden om 'mijn uitvinding' af te maken. De vrije uurtjes na mijn werkdagen bij STEIM spendeerde ik aan het ontwerp van de TimeTosser, zoals het zou gaan heten. Na mijn stage heb ik het geluk gehad om aan de TimeTosser te werken tijdens een ander HKU project.

Wetende dat het uiteindelijke prototype uit twee printplaten met ieder een microcontroller ging bestaan, werd er overeengekomen dat het eindproduct voor het project de firmware voor het moederbord was. Dit was geen makkelijke taak, want om aan te tonen dat de firmware werkte, moest dit natuurlijk wel uitgevoerd kunnen worden. Dit betekende dat het hele moederbord ook moest werken.

Na het project en het studiejaar succesvol te hebben afgesloten, begon in aan het EMMA jaar. Halverwege dit jaar werd het voltooiën van de TimeTosser goedgekeurd als afstudeerproject.

Al deze tijd is de drijfveer achter de TimeTosser mijn eigen enthousiasme geweest. Ik was en ben er van overtuigd dat het TimeTossing principe interessant is voor een hoop mensen. Ook denk ik dat de implementatie in hardware en de manier waarop goede keuzes zijn, de hulp van mede-studente Erika Kiessner is bij het vormgeven van de User-Interface speelt hierin een grote rol.

Zonder daadwerkelijk mensen met dit prototype te laten werken zijn dit natuurlijk aannames en absoluut geen zekerheid. Om tot wat meer zekerheid hierachter te kunnen komen besloot ik mijn thesis toe te wijden aan het onderzoeken van de legitimiteit van mijn aannames en de bruikbaarheid van het TimeTossing principe in het algemeen.

# 1. Inleiding

## 1.1 Toepassingen van geluidssignaalvertraging

Het vertragen van geluidssignalen is een fenomeen waar veel toepassingen gebruik van maken.

Voorbeelden hiervan zijn flanging en chorus. Hierbij wordt een wisselende kleine hoeveelheid vertraging, in de orde van milliseconden, op het signaal toegepast en wordt daarna gemengd met het originele signaal om een kamfilterwerking te krijgen. Doordat de hoeveelheid vertragingstijd wisselt, variëren ook de “tanden” van het kamfilter in frequentie. Dit effect wordt veel toegepast in (gitaar)effectapparatuur.

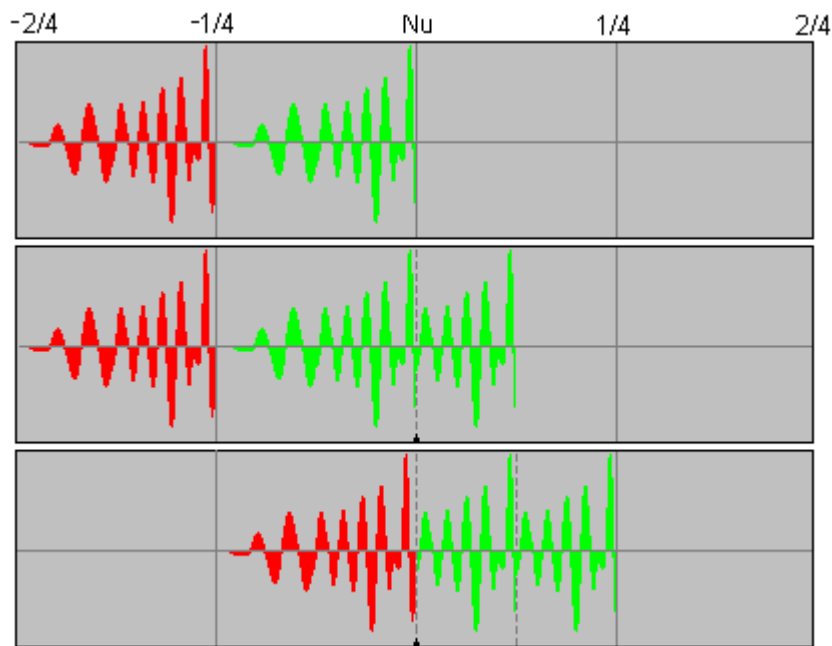
Van de filterwerking die het mengen van een vertraagd signaal met het originele signaal tot gevolg kan hebben, wordt gretig gebruikt gemaakt in digitale audiofilters. Deze vertragingen zijn dan zo klein dat er een middeling van het signaal optreedt. Deze filters vinden toepassing in bijvoorbeeld digitale synthesizers, maar ook in consumentenelektronica zoals CD-spelers.

Een meer voor de hand liggend voorbeeld is het echo-apparaat, waar wederom het originele signaal gemengd wordt met een (verzwakte) vertraagde versie. De vertraging in deze toepassing is vaak in de orde van honderden milliseconden. Door het vertraagde signaal wederom te verzwakken en terug te koppelen naar de vertraging wordt een uitstervende echo bereikt.

Interessanter wordt het wanneer de vertragingstijd gekoppeld is aan het tempo van de muziek die door het echo-apparaat geaffecteerd wordt. De vertragingstijd is dan bijvoorbeeld  $1/8$  noot lang. Hiermee kunnen interessante ritmische effecten worden bereikt.

## 1.2 Het TimeTossing principe

Door de vertragingstijd te variëren en het originele signaal weg te laten, is het mogelijk om delen van het signaal te herhalen. Door de vertragingstijd in ritmische tijdeenheden te kwantiseren kan dit binnen een muzikale context gebruikt worden. Dit principe zal vanaf hier het TimeTossing principe genoemd worden.



Figuur 1. Grafische weergave van het TimeTossing principe. De diagrammen verschillen elk een achtste noot in de tijd. Het bovenste diagram laat het signaal met geen vertraging zien. Een achtste noot later wordt de vertraging een achtste noot gemaakt, waardoor de groene toon wordt herhaald. Hierna wordt de vertragingstijd weer nul, zodat de rode noot klinkt waar deze normaliter ook geklonken zou hebben.

Door de vertragingstijd elke 8e noot een 8e noot groter maken, is het mogelijk om een stukje muziek van 1/8e noot lang eindeloos te herhalen. Natuurlijk is het ook mogelijk om de vertragingstijd niet te laten oplopen, maar te variëren volgens een patroon. Hierdoor wordt effectief de volgorde van het originele signaal gewijzigd.

Nu is het herschikken van audio echter niets nieuws: sinds de uitvinding van de magnetische audiotape is het al mogelijk om een stuk band in stukken op te knippen om deze stukken in een andere volgorde weer aan elkaar te lijmen. Deze techniek werd al in 1952 gebruikt door de *Musique Concrete* pionier Pierre Schaefer in het stuk "Vocalise" (Ernst 1977: 2-3).

Dit arbeidsintensieve proces werd vereenvoudigd met de komst van de eerste digitale samplers. Door de continuerende technologische ontwikkeling is de huidige thuiscomputer in staat deze bewerkingen uit te voeren.

Het TimeTossing principe onderscheidt zich echter van de bovenstaande voorbeelden. Zowel bij de tape als de sampler en de computer moet het signaal eerst worden opgeslagen op een medium, alvorens het bewerkt kan worden. Na het opslaan moet de gebruiker eerst de bewerkingen uitvoeren alvorens het eindresultaat te kunnen beluisteren.

Het TimeTosser principe kent deze limitatie niet. Al gebruikt het principe wel een medium om het signaal te kunnen opslaan, is dit van tijdelijke aard. Het inkomende signaal wordt continu gebufferd en zal na een bepaalde tijd ook weer uit de buffer verdwijnen om vervangen te worden door nieuwe data.

Door de tijdelijke aard van het medium, is het mogelijk om het herschikken in *real-time* te laten plaatsvinden. Immers, de enige parameter die verandert is de vertragingstijd. Het is juist deze eigenschap van het TimeTossing principe wat het in aanmerking laat komen voor gebruik als muziekinstrument.

## **1.3 Het TimeTosser prototype**

### **1.3.1 Implementatiecriteria**

Om het TimeTossing principe te implementeren in een zelfstandig opererend stuk hardware zijn een aantal elementen noodzakelijk voor het functioneren. Mede door de lage prijs van digitale elektronica en de relatieve simpliciteit ervan is gekozen om digitaal geheugen te gebruiken als tijdelijk opslagmedium.

Gezien de digitale aard van het geheugen is het noodzakelijk om het ingaande audiosignaal eerst te digitaliseren, met behulp van een analoog naar digitaal omzetter (ADC).

Ook moet het digitale audiosignaal weer geconverteerd worden naar een analoog signaal alvorens het door mensen te beluisteren is. Dit gebeurt met een digitaal naar analoog omzetter (DAC). Doordat er gekozen is om het prototype zo zelfstandig mogelijk te kunnen laten functioneren zijn de ADC en DAC opgenomen in het ontwerp.



Een ander vereiste van het TimeTossing principe is dat de vertragingstijd *real-time* ingesteld moet kunnen worden. Gezien de beoogde muzikale context van het prototype, is er voor gekozen om de vertragingstijd in stappen te kunnen instellen. Deze stappen zijn veelvoud van nootlengtes, variërend van nul tot zeven stappen. Ook is er gekozen om deze stapgrootte instelbaar te maken van  $\frac{1}{4}$  noot tot  $\frac{1}{6}$  noot.

Het aantal stappen en instelbare stapgroottes is gekozen volgens de aanname dat het prototype vooral gebruikt wordt met muziek in vierkwartsmaat, waar de meeste ritmische informatie zich binnen twee maten afspeelt.

Ook is de aanname gemaakt dat de kleinste ritmische eenheid in deze muziek een zestiende noot is. Hieruit volgt dat de kleinst mogelijke vertraging één zestiende noot is, de grootste zeven kwartnoten. Om het herschikken in triolen mogelijk te maken zijn ook stapgroottes van  $\frac{1}{6}$  en  $\frac{1}{12}$  noot mogelijk.

### **1.3.2 Ontwerpkeuzes**

Door het gebruik van muzikale tijdeenheden is de enige parameter, namelijk de vertragingstijd, ineens onderverdeeld in drie parameters: tempo, stapgrootte en de veelvoud hiervan.

Er is gekozen om elke parameter zijn eigen groep drukknoppen te geven op het bedieningspaneel. Dit heeft als voordeel dat de juiste parameter snel en trefzeker kan worden ingesteld. Dit in tegenstelling tot bijvoorbeeld een meerstandendraaiknop, waar trefzekerheid wordt opgeofferd door snelheid.

Voor het tempo wordt echter één enkele drukknop gebruikt. Het herhaaldelijk en ritmisch indrukken van deze knop zorgt voor de berekening van het tempo: de gemiddelde tijd tussen het indrukken wordt geïnterpreteerd als de lengte van één kwartnoot.

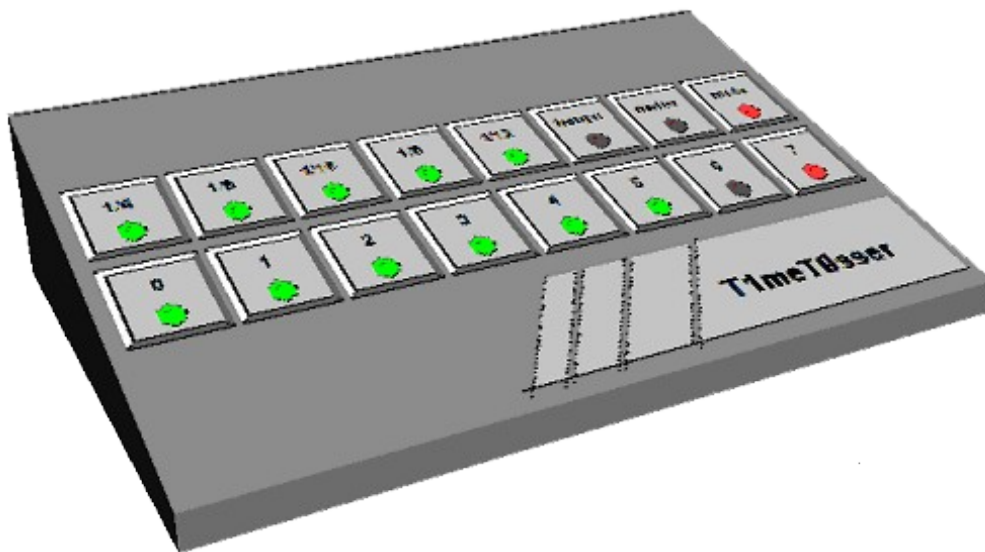
Deze methode is gekozen aangezien een tempo analyse op het audiosignaal een rekenkundig complex algoritme vergt, wat bovendien niet altijd even betrouwbaar is (Alonso, M., David, B., Richard G., 2005). Veel mensen daarentegen, zijn zeer goed in staat het tempo van muziek te herkennen (Dixon, S., 1997). Een bijkomend voordeel van een menselijk ingegeven tempo is dat het geïnterpreteerde tempo niet per se het tempo van de ingaande muziek hoeft te zijn, wat andere stapgroottes mogelijk maakt.

Aangezien de gebruikte microcontrollers niet over voldoende vrije rekenkracht beschikken om een dergelijk algoritme te kunnen toepassen, maar wel over een timermodule, en er vanuit gegaan kan worden dat de gebruiker over voldoende muzikale capaciteiten beschikt om het tempo uit de muziek te extraheren, is besloten het tempo door de gebruiker te laten ingeven.

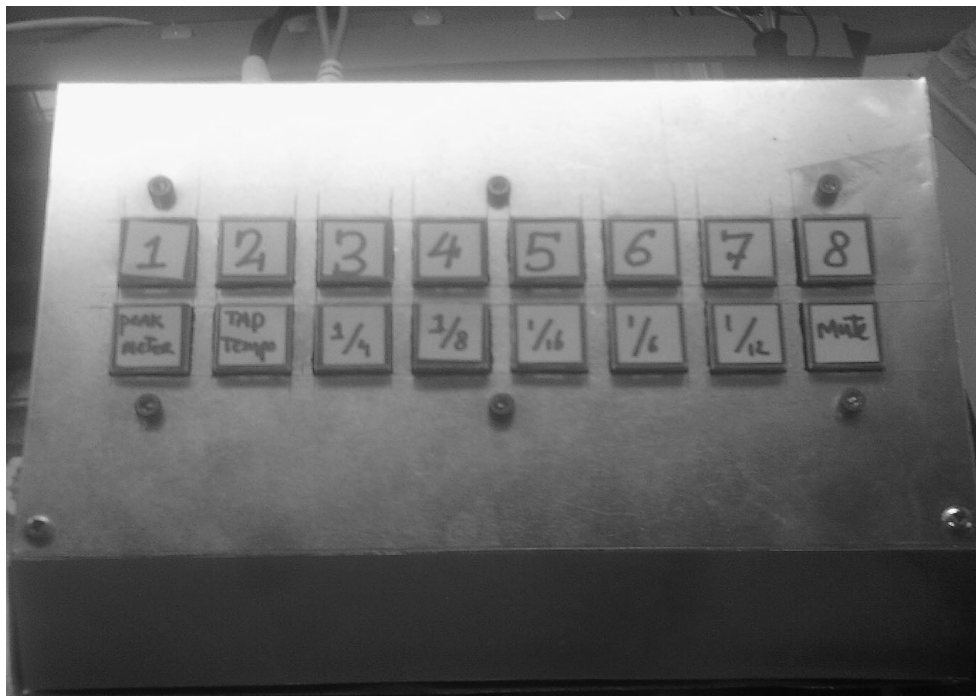
Naast de acht knoppen om de veelvoud van de stapgrootte in te stellen, de vijf knoppen om de stapgrootte in te stellen en de tempo knop, zijn er nog twee andere knoppen in het leven geroepen. De ene schakelt het geluid uit, dit wordt weer ingeschakeld door nog een keer op deze of elke andere knop te drukken. De andere overige knop activeert de piekmeter. Deze is in het leven geroepen om de amplitude van het ingangssignaal af te regelen, zodat er zo min mogelijk vervorming optreedt.

De piekmeter wordt gevisualiseerd door LEDs in de drukknoppen. De zestien drukknoppen van het bedieningspaneel zijn namelijk in twee rijen van acht knoppen gerangschikt, zoals te zien in figuur 2. De bovenste rij knoppen vertegenwoordigt het linker kanaal, de andere het rechter.

In de piekmeter modus lichten de LEDs groen op, met uitzondering van de twee meest rechter LEDs. Deze rood oplichtende LEDs lichten alleen op als de amplitude van het ingangssignaal te groot is, wat tot vervorming leidt. Wanneer de piekmeter modus niet actief is lichten de LEDs rood op, afhankelijk van de gekozen stapveelvoud en stapgrootte.



Figuur 2. Prototype schets. Onderaan de knoppen voor de veelvoud van de stapgrootte, Linksboven de knoppen voor de stapgrootte. De piekmeter is actief.



Figuur 3. Het prototype. De LED-indicatoren bevinden zich onder de papieren knop-inzetten. De stap-veelvoud knoppen van rij gewisseld om ze beter toegankelijk te maken

## **1.3 Relevantie en vraagstelling**

### **1.3.1 Formulering van hoofd-onderzoeksvraag**

Bij het ontwerp van het TimeTosser prototype zijn een aantal aannames gemaakt. Allereerst is er vanuit gegaan dat het TimeTossing principe het best tot zijn recht komt in hardware. Dit is gebaseerd op aannames over de doelgroep, alsmede over de praktische toepassing van het principe, namelijk dat het TimeTossing principe ingezet kan worden als performance tool / muziekinstrument voor gebruik door DJ's en muzikanten.

Om de inzetbaarheid en daarmee de validiteit van het TimeTossing principe aan te tonen, zullen de gemaakte aannames bij het ontwerp van het prototype onderzocht dienen te worden. Een vergelijking tussen vergelijkbare bestaande commerciële producten, alsmede voorbeelden van bestaande vergelijkbare fenomenen in muziek zou een meer impliciet bewijs kunnen leveren voor de relevantie van het TimeTosser principe. Hieruit volgt de volgende onderzoeksvraag:

**In hoeverre zou het TimeTosser principe vernieuwend in het gebruik als performance-tool / muziekinstrument kunnen zijn?**

### **1.3.2 Formulering van sub-onderzoeksvragen**

Het onderzoek naar de relevantie van het TimeTossing principe is op te delen in een deel literatuuronderzoek en een deel kwalitatief onderzoek. In het literatuuronderzoek zullen zoals gezegd vergelijkingen getrokken worden tussen bestaande muzikale fenomenen. Dit zal kunnen uitwijzen of de aannames over de doelgroep gerechtvaardigd zijn. Dit wordt in hoofdstuk 2 behandeld. In dit onderzoek worden de volgende subvragen onderzocht:

- **Hoe verhoudt het TimeTosser principe zich tot de principes gebruikt in bestaande commerciële producten?**
- **Welke muzikale fenomenen laten zich vergelijken met het TimeTosser principe?**
- **Welke doelgroep heeft zou baat kunnen hebben bij het TimeTosser principe?**

Hoofdstuk 3 is gewijd aan het onderzoeken van de aannames die zijn gebruikt bij het ontwerp van het prototype door middel van kwalitatief onderzoek. Dit onderzoek wordt

verricht door proefpersonen uit de doel groep te laten werken met het prototype en deze daarna een vragenlijst te laten invullen. Dit kwalitatief onderzoek kan de validiteit van de gemaakte aannames aantonen.

De volgende sub-onderzoeksvragen worden onderzocht in dit hoofdstuk:

- **In hoeverre kan het TimeTosser-principe bruikbaar zijn voor de beoogde doelgroep?**
- **In hoeverre zijn de gemaakte aannames bij het ontwerp van het TimeTosser prototype legitiem?**

## **2 Literatuuronderzoek**

### **2.1 Methode van onderzoek**

Het literatuuronderzoek bestaat uit twee delen. Het doel van het eerste deel is om vergelijkbare fenomenen in de muziek te vinden. Door de verschillen en overeenkomsten van deze fenomenen te onderzoeken wordt duidelijk of het TimeTosser principe een bruikbare muzikale toepassing zou kunnen hebben. Ook zouden deze fenomenen al dan niet de gebruikersgroep kunnen bevestigen, zoals beschreven in hoofdstuk 1.

In het tweede deel van het literatuuronderzoek wordt gezocht naar bestaande commerciële producten die vergelijkbare eigenschappen hebben als aanwezig in het TimeTosser principe. Ook hier worden de verschillen en overeenkomsten onderzocht, hopelijk aantonend dat het TimeTosser principe een uniek verschijnsel is.

### **2.2 Vergelijkbare bestaande oplossingen**

#### **2.2.1 Resultaten**

##### **2.2.1.1 Hardware**

###### **2.2.1.1.1 Samplers**

Met de komst van de Fairlight CMI, de eerste commercieel verkrijgbare digitale samplers, was het al mogelijk de gesampelde data te bewerken en in principe op deze manier de volgorde hiervan te wijzigen (Carlos, Stewart, 1985).

Dit proces is echter behoorlijk arbeidsintensief: het opzoeken van splitspunten en daadwerkelijk knippen van de samples gebeurde handmatig en moest vaak visuele ondersteuning ontberen.

Sommige samplers beschikken over een meer gebruikersvriendelijkere manier om samples op te knippen. Een voorbeeld hiervan is loop-dividend functie van de Yamaha

A4000 en A5000 sampler. Hier geeft de gebruiker aan in hoeveel stukken de sample opgeknipt moet worden. Hierna worden deze losse stukjes automatisch toegewezen aan MIDI-nootnummers, zodat deze bespeeld kunnen worden (Yamaha 1999: 221).

Een andere functie op de A4000 / A5000 is loop-remix. Deze functie stelt de gebruiker een sampler op te delen in gelijke stukken, waarna deze in willekeurige volgorde gerangschikt worden. Ook kunnen deze stukken onderling op willekeurige wijze in toonhoogte en afspeelrichting gewijzigd worden (Yamaha 1999: 125). Alhoewel deze functie een spectrum aan mogelijkheden biedt, heeft de willekeurige wijze waarop deze bewerking plaatsvinden als resultaat dat de gebruiker geen totale controle over het proces heeft.

In beide functies wordt de sample in gehele stukken verdeeld. Een nadeel hiervan dat individuele componenten van de sample exact in de maat moeten zitten: als de sample een "swing" feel bezit, zal dit resulteren in dat de aanslag van sommige noten zich net voor of net na het knippunt bevindt.

De Beat Munger, een functie van het besturingssysteem dat wordt gebruikt in de Ultra lijn van E-mu samplers, is vergelijkbaar met de loop-remix en loop-divide functies uit de Yamaha samplers. Deze functie is echter intelligenter in de zin dat het de loop analyseert naar de daadwerkelijke losse componenten.

De losse beats in deze loop kunnen op het display van de sampler getoond worden, waarna deze aan en uit kunnen worden gezet, wat effectief de maatsoort van de loop verandert. Doordat de sampler zich bewust is van de losse onderdelen van de loop kan, naast de maatsoort, het tempo en zelfs *de feel* van de loop worden aangepast zonder dat er artefacten optreden zoals bij time-stretching algoritmes (Burger, 2000).

Al biedt de Beat-Munge functie de mogelijkheid om loops qua tempo, *feel* en maatsoort te veranderen, is het niet direct in staat om de volgorde van het signaal te veranderen, in tegenstelling tot de bovengenoemde Yamaha samplers.

Wél kan de verkregen loop later alsnog handmatig in stukken worden geknipt, waarna deze toegewezen kunnen worden aan MIDI-nootnummers, wat het bespelen en dus het wijzigen van de originele volgorde van de loop in real-time mogelijk maakt.

AI bieden samplers de mogelijkheid om de volgorde van een stuk audio te wijzigen, zijn er toch enkele verschillen ten opzichte van het TimeTosser Principe.

Als eerste dient de audio eerst opgeslagen te worden in het geheugen van de sampler, voordat deze bewerkt kan worden. In TimeTosser kunnen deze processen simultaan uitgevoerd worden.

Ten tweede dienen voor een ritmisch resultaat de opgeknipte stukken audio in de maat getriggered te worden. Bij het TimeTosser principe is dit niet per se een vereiste: doordat de onderlinge vertragingstijden ritmisch gekwantiseerd zijn, zal het schakelen hiertussen nooit een a-ritmisch resultaat tot gevolg hebben.

### **2.2.1.1.2 Echo-apparaten**

Aangezien het TimeTosser principe, evenals het principe achter echo-apparaten, berust op het vertragen van audio, kunnen echo-apparaten in theorie gebruikt worden om het TimeTosser-principe te emuleren.

Tot de parameters van een conventioneel echo-apparaat behoren de ingestelde vertragingstijd, de verhouding tussen vertraagd en onvertraagd signaal, en de hoeveelheid terugkoppeling van het vertraagde signaal naar het begin van de vertragingstijd.

Wanneer de hoeveelheid terugkoppeling nul gemaakt wordt en er geen onvertraagd signaal meer aanwezig is in het uitgangssignaal, wordt, door de vertragingstijd te variëren het, mogelijk om het TimeTosser principe te evenaren.

Om het principe succesvol te emuleren, moet de vertragingstijd zeer direct en accuraat en bovendien ritmisch gekwantiseerd ingesteld kunnen worden. Dit is bij echo-apparaten niet het geval: vaak moet de vertragingstijd ingesteld worden met behulp van een draaiknop. Deze methode biedt weinig nauwkeurigheid, behalve dan wanneer er visuele ondersteuning is aan de hand van een display. Deze methode is lang niet snel genoeg om de snelle variaties in vertragingstijd die benodigd zijn bij herschikken van muziek in complexe patronen te accommoderen.

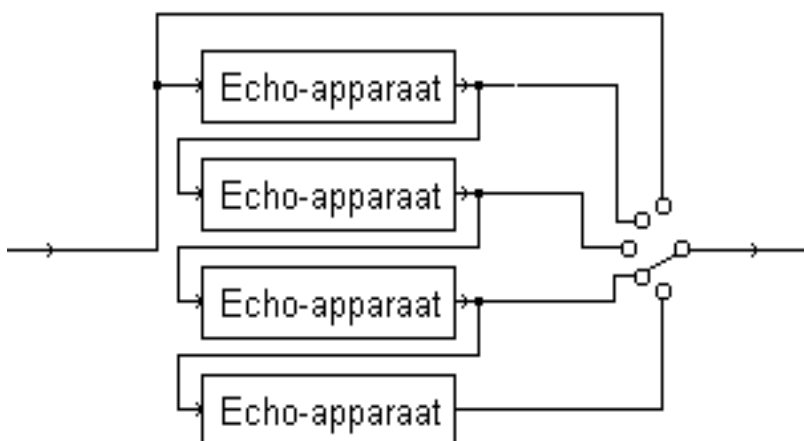
Een andere methode om de vertragingstijd in te stellen, zoals aanwezig bij de TC-electronic D Two, maakt gebruik van tap-tempo: het tijdsinterval tussen het meervoudig indrukken van een knop. Ook deze methode is niet snel genoeg om snel opeenvolgende

sprongen in de vertragingstijd te bewerkstelligen.

Een andere methode om snelle ritmisch gekwantiseerde veranderingen in de vertragingstijd te maken kan worden bereikt door meerdere echo-apparaten in te stellen op dezelfde vertragingstijd en deze vervolgens achter elkaar te schakelen: ieder apparaat, met uitzondering van het laatste, is met zijn uitgang verbonden aan de ingang van het volgende apparaat. Hetingangssignaal wordt dan verbonden met de ingang van het eerste apparaat. Door het signaal tussen elk apparaat af te tappen en vervolgens tussen het originele onvertraagde signaal en de afgetapte vertraagde signalen te schakelen, kan men het signaal in veelvouden van de ingestelde tijd vertragen. Essentieel hierbij is dat het uitgangssignaal van elk echo-apparaat niet gemengd is met het originele signaal.

Voor het schakelen tussen signalen zijn verschillende oplossingen mogelijk. Een daarvan is de zogenaamde audio-switcher, zoals bijvoorbeeld de AS4x1 van Burst Electronics. Dit apparaat gebruikt drukknoppen om één van de aangesloten bronnen door te sluisen naar de uitgang.

Alhoewel deze oplossing het TimeTossing principe kan emuleren, biedt het niet de snelle bediening zoals bij het TimeTosser prototype, zoals beschreven in hoofdstuk 1.2. Hiernaast maken de noodzakelijkheid van meerdere echo-apparaten en de bijbehorende hoeveelheid aansluiten deze oplossing een weinig praktische implementatie.



Figuur 4. Schematische weergave van het gebruik van vier echo apparaten voor het emuleren van het TimeTossing principe. Elk echo-apparaat is ingesteld op dezelfde vertragingstijd. Door de positie van de schakelaar zal de vertragingstijd bij het uitgangssignaal in dit geval drie maal de ingestelde vertragingstijd zijn



De Headrush, gefabriceerd door Akai, is een zogenaamde multi-tap delay. In tegenstelling tot conventionele echo-apparaten, is een multi-tap delay in staat om verscheidene echo's van verschillende lengtes aan het originele signaal toe te voegen door het signaal in verschillende stadia in de vertragingsslijn af te tappen.

Een eigenschap van de Headrush is dat deze echo's, waarvan de onderlinge vertragingstijden gelijk kunnen zijn, elk naar een afzonderlijke uitgang kunnen worden gestuurd. Aangezien de Headrush beschikt over vier "taps" kan door gebruik te maken van de afzonderlijke uitgangen één enkele Headrush de vier echo-apparaten uit figuur 4 vervangen.

De vertragingstijd tussen elke tap kan ingesteld worden met behulp van tap-tempo, wat deze oplossing redelijk direct maakt.

De combinatie van de Headrush samen met een audio-switcher zorgt voor een relatief eenvoudig systeem, wat in staat is om het TimeTosser principe te emuleren. Het is echter geen integrale oplossing aangezien deze configuratie twee apparaten en een aanzienlijke hoeveelheid bekabeling vergt. Hiernaast zijn er maar een beperkt aantal vertragingstijden mogelijk, de Headrush beschikt immers over vier "taps".

Uitbreiding van dit systeem om het aantal mogelijke vertragingstijden te vergroten is mogelijk door een tweede Headrush aan te sluiten, echter vergroot dit de complexiteit van het systeem aanzienlijk. Een uitbreiding stelt verdere eisen aan de audio-switcher, aangezien er nu meerdere signalen zijn om tussen te schakelen. Hiernaast wordt het nu noodzakelijk om de vertragingstijd op beide apparaten in te stellen. Gezien de aard van deze procedure is de kans vrij klein dat beide apparaten exact dezelfde vertragingstijd worden ingesteld.

## **2.2.1.2 Software**

### **2.2.1.2.1 *Beat-Slicers***

In de zomer van 1994 brengt het toen net opgerichte Zweedse softwarebedrijf Propellerhead het programma Recycle uit (Kurasaki, 2005). Dit programma stelt de gebruiker in staat om een audio bestand eenvoudig op te splitsen in *slices*. Een drumloop kan op deze manier geautomatiseerd in stukken worden geknipt, enigszins vergelijkbaar met de hierboven beschreven functies van E-MU en Yamaha samplers.

Deze slices kunnen direct naar hardware samplers worden geëxporteerd. Ook kan een midi-file worden geëxporteerd, wat gebruikt kan worden om de sampler aan te sturen.

Deze benadering maakt het mogelijk om de groove van de sample te veranderen, evenals het tempo waarop deze wordt afgespeeld.

Doordat Propellerhead nauwe banden onderhield met Steinberg (Kurasaki, 2005), onder andere bekend van het populaire sequencing / editing pakket CubasCubase, wordt het snel daarna mogelijk om de gemanipuleerde samples, opgeslagen in het .REX bestandsformaat, te importeren in dit pakket.

Op het principe van Recycle wordt tegenwoordig nog steeds verder geborduurd. Recente voorbeelden hiervan zijn Synchronic van Digidesign en LiveSlice van LiveLab. Ook is er een gratis look-alike van Recycle voor Linux: Freecycle.

Alhoewel deze *beat-slicers* in staat zijn om audio te herstructureren, moet net als bij het traditioneel bewerken van audio deze eerst op een medium zijn opgeslagen voordat deze bewerkt kan worden.

Ook al bieden sommige *beat-slicers real-time* controle en is het in sommige gevallen zelfs mogelijk om de losse slices te triggeren en daardoor te bespelen zijn als een instrument, ligt het achterliggende principe dichterbij de sampler dan het TimeTosser principe.

#### **2.2.1.2.2 Supatrigga**

De gratis VST plug-in Supatrigga van Smartelectronix maakt het mogelijk om binnenkomende audio automatisch op te delen in stukjes, waarna deze stukjes op willekeurige wijze afgespeeld worden. De wijze waarop dit gebeurt is in te stellen volgens een aantal parameters, waarvoor individueel de kans dat deze parameters invloed op het signaal uitoefenen ingesteld kan worden. Supatrigga kan de stukjes herhalen, de afspeelsnelheid tot nul laten afnemen, achterstevoren afspelen en herschikken in de tijd. Het voordeel van supatrigga is dat het net als het TimeTosser berust op een buffer, waardoor het niet noodzakelijk is de audio op te slaan alvorens deze te kunnen bewerken.

Alhoewel er met Supatrigga resultaten bereikt kunnen worden die identiek klinken als wat er met het TimeTosser principe mogelijk is, is de manier van bedienen compleet verschillend. Alhoewel er sprake is van real-time controle, is dit over de parameters alleen. Het eindresultaat berust puur op kans, dit is wezenlijk verschillend van de absolute controle die de gebruiker heeft bij het TimeTosser principe.

Hiernaast beschikt Supatrigga niet over zijn eigen tempo: dit wordt verkregen door de VST host. Een voordeel hiervan is dat het mogelijk is om tempo-analyses uit te voeren als de host hiertoe in staat is. Ook zijn er VST hosts, zoals Ableton Live (Sasso 2003) die over tap-tempo functionaliteit beschikken.

### **2.2.1.2.2 Modulaire audio softwarepakketten**

Sinds thuiscomputers halverwege de jaren negentig krachtig genoeg werden, ontstonden er platforms waarmee in real-time audio signalen bewerkt en opgewekt kunnen worden. Voor een aantal van deze pakketten, zoals Native Instrument's Reaktor en Cycling 74's Max/MSP, moet worden betaald. Er zijn ook soortgelijke platforms gratis verkrijgbaar. Een voorbeeld hiervan is PureData, ontworpen door Miller Puckette, een van de ontwikkelaars van Max/MSP.

Al deze bovengenoemde platforms maken het mogelijk om verscheidene modules aan elkaar te koppelen, vergelijkbaar met de modulaire analoge synthesizers uit de jaren '60 en '70. Deze platforms beschikken niet alleen over de klassieke synthesesmodules zoals sequencers, filters, oscillatoren, envelope-generatoren en mixers, maar ook modules op een kleiner functioneel niveau.

Met deze pakketten is het zeer goed mogelijk om de exacte functionaliteit van het TimeTosser principe te realiseren. Het is zelfs mogelijk om het TimeTosser prototype zoals beschreven in hoofdstuk 1.3 exact te simuleren, compleet met visuele indicatoren en piekmeter.

Door de ondersteuning van het MIDI protocol is mogelijk om deze pakketten in real-time te bedienen. Door elke afzonderlijke functie van het TimeTosser prototype te koppelen aan MIDI nootnummers, is het mogelijk om een perfect bespeelbare simulatie van het TimeTosser prototype te creëren.

### **2.2.3 Conclusie**

Vandaag de dag zijn er tal van oplossingen, zowel in hardware als in software, waarmee vergelijkbare resultaten kunnen worden bereikt als met het TimeTosser principe mogelijk is.

De onderzochte oplossingen verschillen vaak in het achterliggende principe, zoals bij de sampler en de beat-slicer software. Bij andere oplossingen, zoals bij de Akai Headrush en Suppatrigga, zit het verschil in de controle en bespeelbaarheid.

Zowel de sampler, de beat-slicer software en supatrigga zijn bewust ontworpen om deze resultaten te behalen. De Headrush is duidelijk niet ontworpen met deze resultaten als doel, dit blijkt ook uit de benodigde extra apparatuur en de complexiteit van het systeem dat dit teweeg brengt.

De onderzochte modulaire audio software pakketten zijn perfect in staat het TimeTosser principe te kunnen realiseren. Deze producten zijn echter niet ontworpen voor een enkele functie: het is juist het spectrum aan mogelijkheden wat er toe leidt dat het TimeTosser principe realiseerd kan worden.

## Literatuurlijst

Alonso, M., David, B., Richard G. (2005) *Tempo extraction in audio recordings*

Burger, J. (2000) "E-MU Systems Ultra" *Electronic Musician*, URL bezocht op 14 Augustus 2007

Carlos, M., Stewart, T. (1985) *Fairlight CMI v4 \*\* 13 Operation Manual*. Sydney: Fairlight Instruments Pty. Ltd.

Dixon, S. (1997) *Beat Induction and Rhythm Recognition*

Ernst, D. (1977) *The Evolution of Electronic Music*. New York: Schirmer Books

Kurasaki, K. (2005) "History of Rebirth Development" *The Rebirth Museum*, URL bezocht op 13 Augustus 2007

Ranck, J. (2007) "Classified Hip-Hop" *Simmons College*, URL bezocht op 4 Augustus 2007

Sasso, L (2003) "Ableton Live 2.0" *Electronic Musician*, URL bezocht op 14 Augustus 2007

White, Paul (1996) "Sampling basics: part 1" *Sound on Sound*, URL bezocht op 1 Augustus 2007

(1999) *Professional Sampler A5000 / A4000 Operation Manual*, Hamamatsu: Yamaha Corporation, Pro Audio & Digital Musical Instrument Division