

# Wave Terrain Synthese

Hendrik Dinger, 21.05.2007

# Geschichtlicher Hintergrund

John Bischoff, Rich Gold und Jim Horton:

„A microcomputer-based network for live performance“  
(Computer Music Journal, 1978)

=> überarbeitet und zusammengefasst von Curtis Roads und John Strawn:  
„Foundations of Computer Music“ (the MIT-Press, 1985)

Yasuhiro Mitsuhashi:

„Audio signal synthesis by functions of two variables“  
(„Journal of the Audio Engineering Society“ ,1982)

A. Borgonovo und G. Haus:

„Musical sound synthesis by means of two-variable functions:  
experimental criteria and results “  
(International Computer Music Conference, 1984/86)

# Mathematik und Grundlagen der WTS

Mit Hilfe der Wave Terrain Synthese lassen sich, durch Abtastung dreidimensionaler Funktionstabellen zweidimensionale Wellenformen erzeugen.

Die Achsen der 3D-Tabelle sind:

x-Achse	(links/rechts)
y-Achse	(vorne/hinten)
z-Achse	(unten/oben)

## Wave Terrain („Wellenfläche“)

- eine Funktion mit 2 Variablen,  $f(x,y)$ , erzeugt eine 3D-Oberfläche, bei der sich die Z-Achse in jedem Punkt aus den Werten der xy-Paare ergibt und die Wellenform der 3D-Oberfläche bestimmt.

(„two-variable function synthesis“, Borgonovo u. Haus, 1986)

Eine solche 2-Variablen-Funktion kann durch die Multiplikation zweier Funktionen mit jeweils einer Variablen errechnet werden:

$$f(x,y)=f(x)*f(y)$$

- um die resultierende Wellenform vorweg bestimmen zu können, muss für x und y ein Wertebereich bestimmt werden:

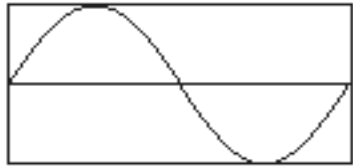
$$f(x,y)=f(x)*f(y) \quad ; \text{mit } x \text{ und } y \text{ in } [0,1] \text{ od. } [-1,1], \text{ etc.}$$

- 2 Voraussetzungen gelten:

1. Beide Funktionen  $f(x)$  und  $f(y)$  und deren erste Ableitung müssen einen kontinuierlichen Verlauf vorweisen, innerhalb des Terrains.
2. Beide Funktionen  $f(x)$  und  $f(y)$  müssen an ihren Enden Null sein

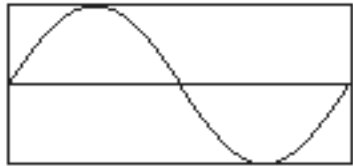
Beispiele für Wave Terrains:

x axis: ftable 2

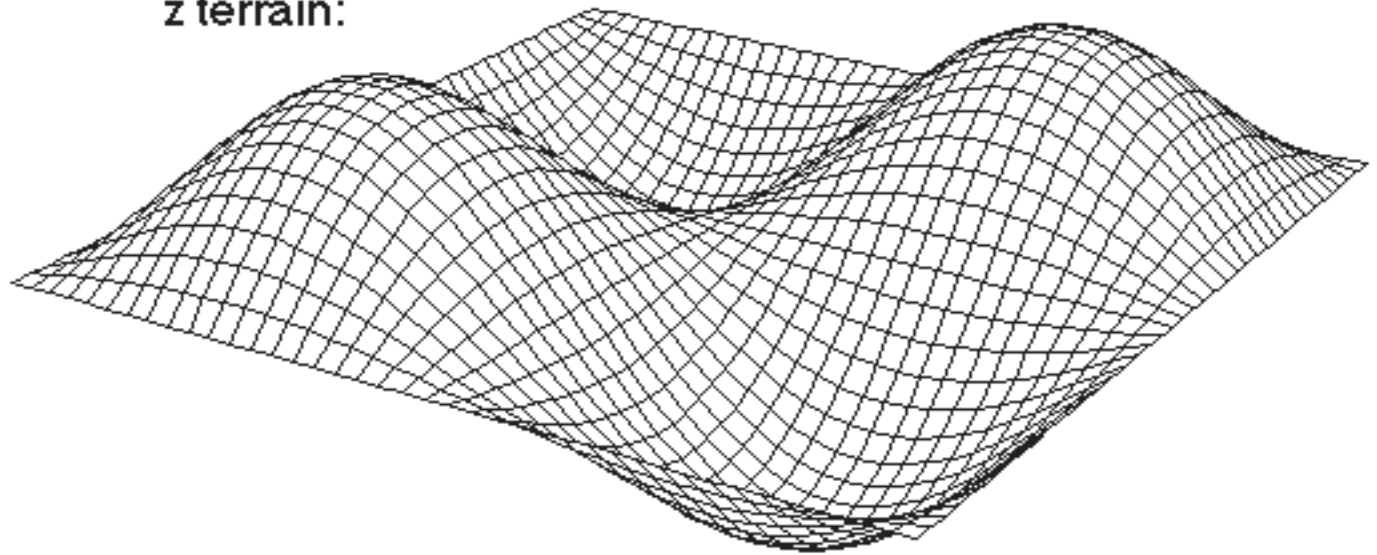


$x$

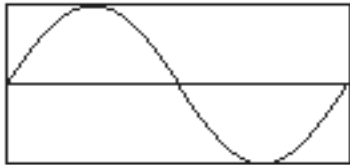
y axis: ftable 2



z terrain:

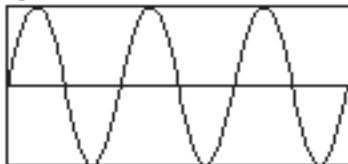


x axis: ftable 2

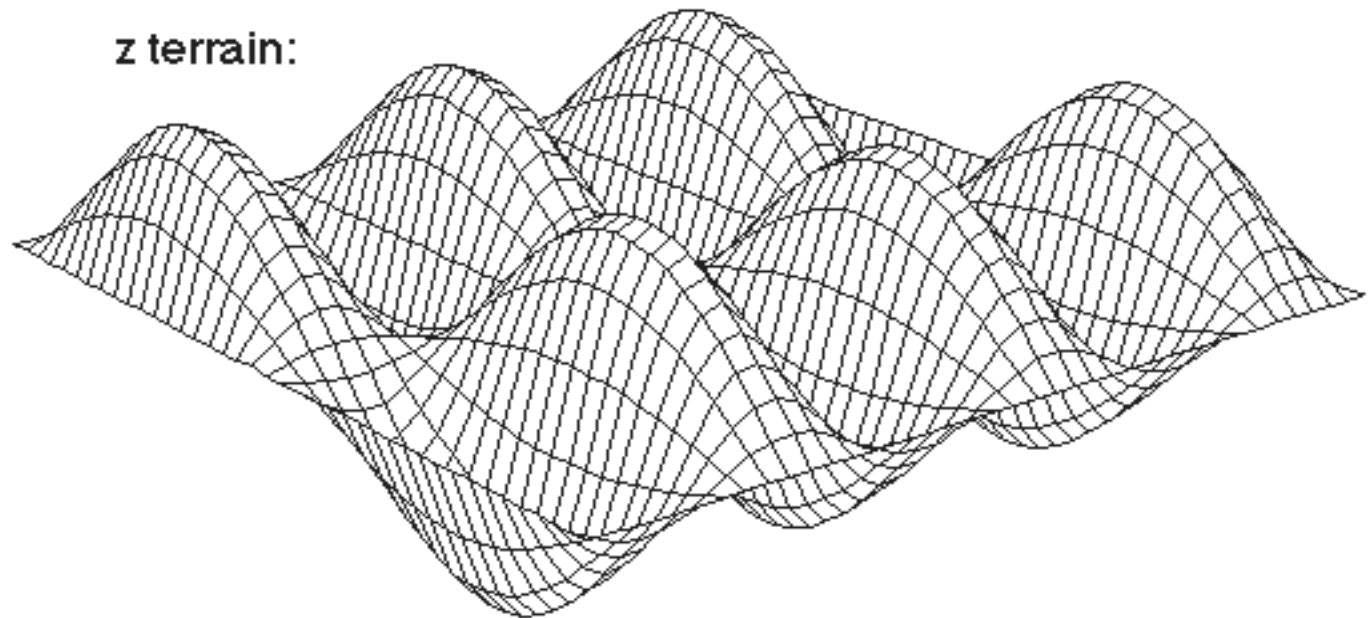


$x$

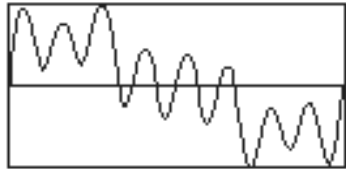
y axis: ftable 61



z terrain:

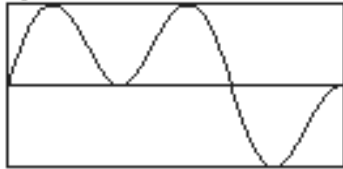


x axis: ftable 62

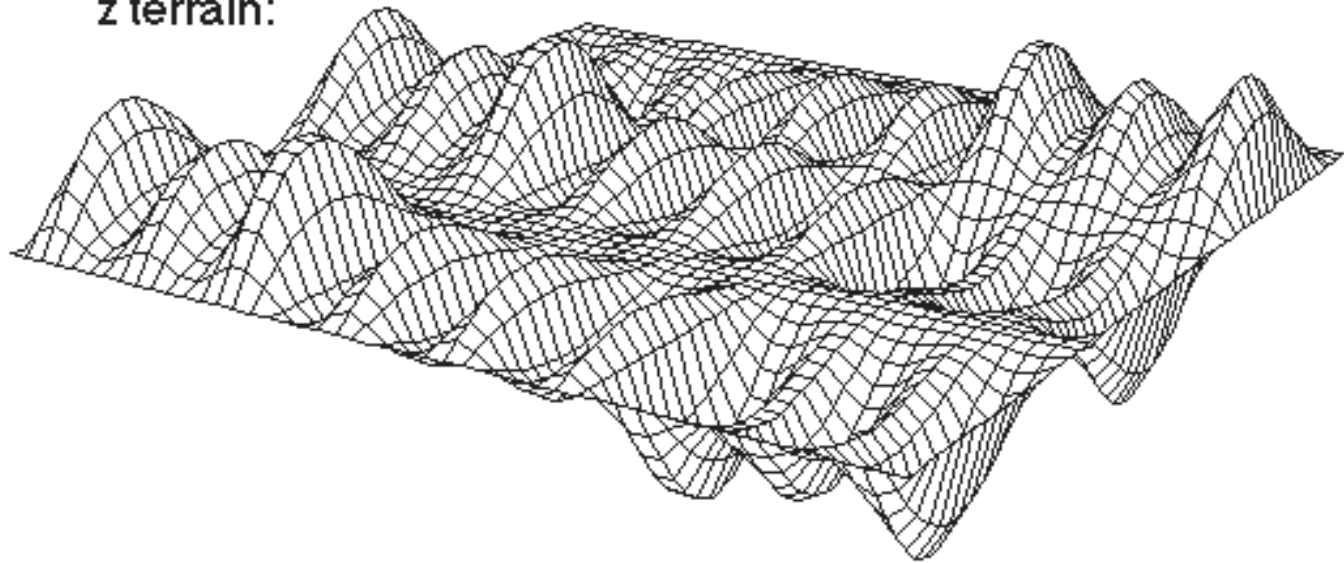


$x$

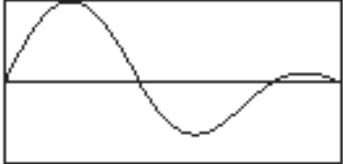
y axis: ftable 10



z terrain:

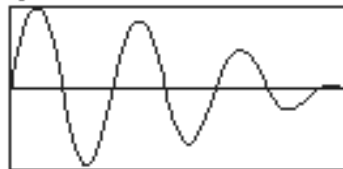


x axis: ftable 63

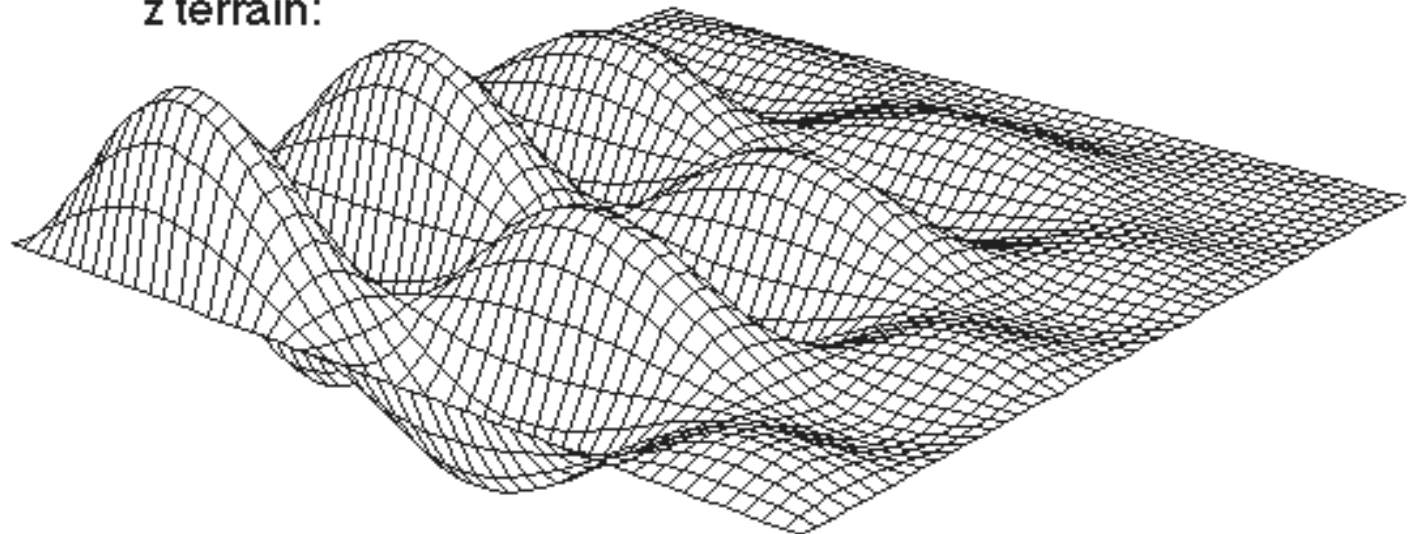


$x$

y axis: ftable 64



z terrain:



**Die resultierende Wellenform ist von der Wellenform des Wave Terrains und der Verlaufsbahn der Abtastung („Orbit“) abhängig**

## **Orbits („Umlaufbahn“, astro.)**

- Als Orbit wird der Bewegungsablauf der Abtastung eines Wave terrains bezeichnet
- Der Orbit kann eine gerade oder gebogene Linie quer über das Wave Terrain sein, oder aber auch eine Funktion, sinusoid oder elliptisch, etc.. Der Orbit wird mathematisch in die Funktion des Wave Terrains eingebunden.

Bsp.: Orbit als Funktion :

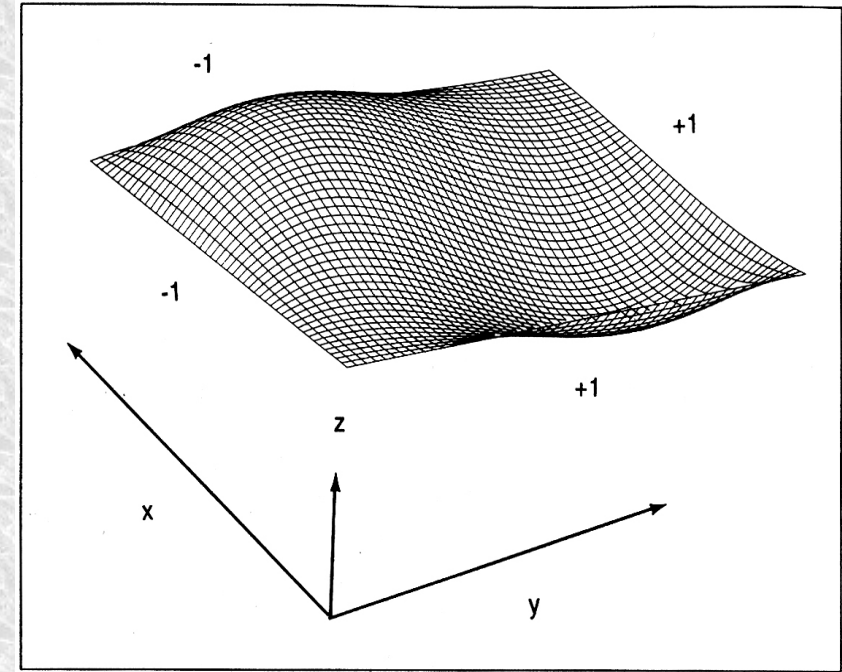
$$x=g(t), y=h(t)$$

$$\Rightarrow f(x,y) = f(g(t)) * f(h(t))$$

Fixiertes Wave Terrain, periodischer Orbit:

Bsp.:  
Wave Terrain:

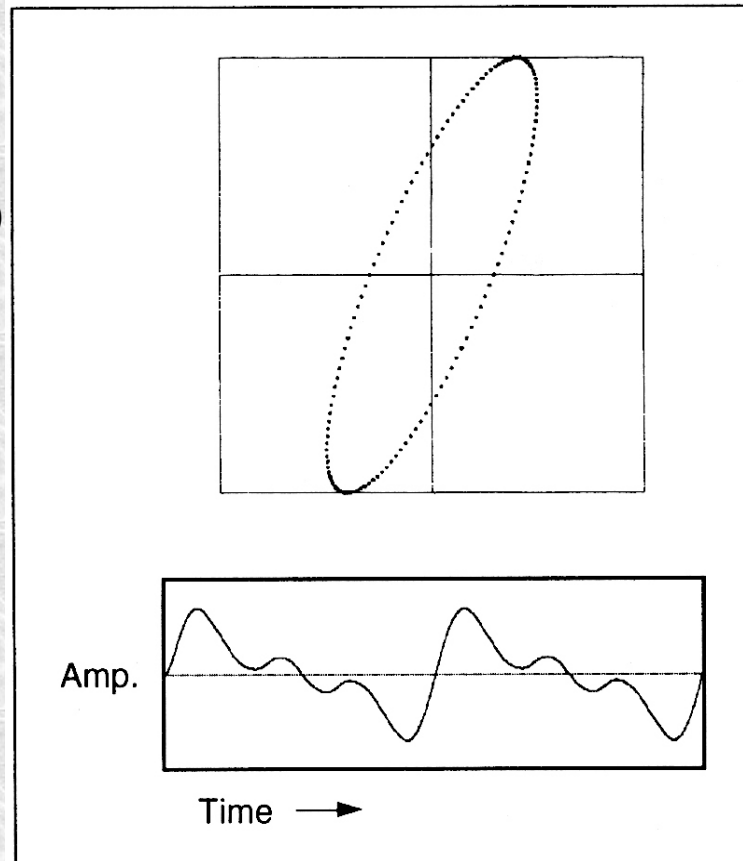
$$z(x,y) = (x-y)*(x-1)*(x+1)*(y-1)*(y+1) \text{ mit } x \text{ und } y \text{ in } [-1,1] \Rightarrow$$



Elliptischer Orbit:

$$x = 0.5 * \sin(8\pi t + \pi/5)$$
$$y = \sin(8\pi t)$$

=>



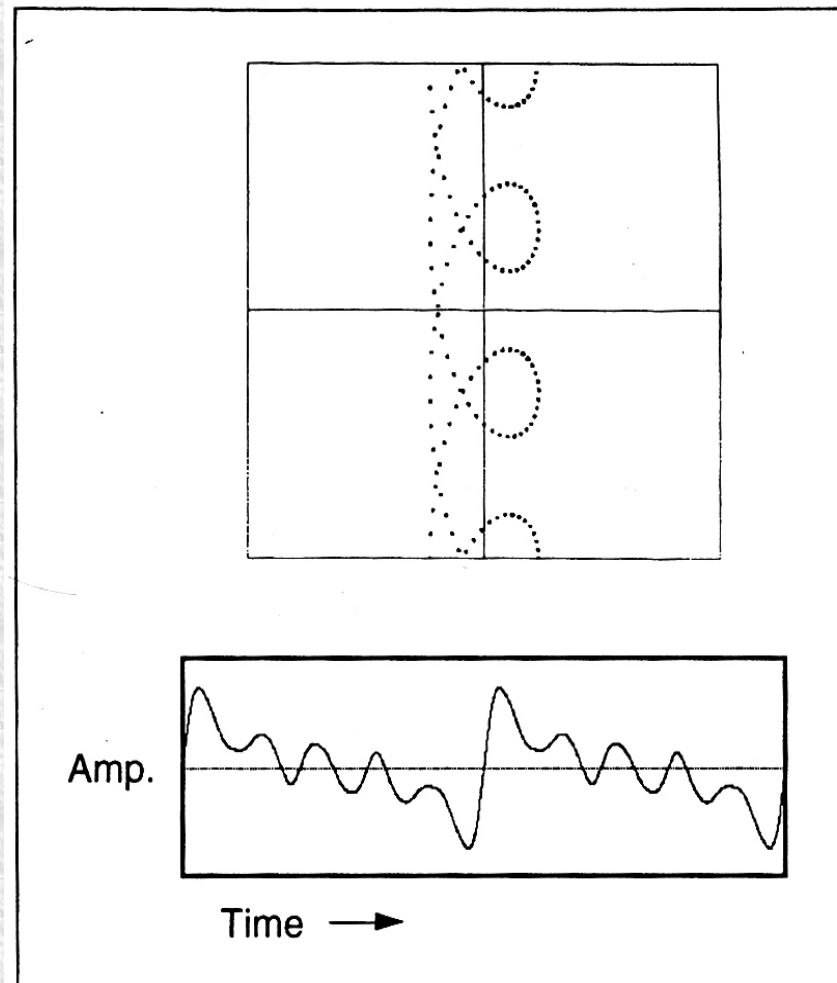
<= Resultierende Wellenform

Periodischer „looping“ Orbit, selbes Wave Terrain:

$$x=0.23*\sin(24\pi t)$$

$$y=(16*t)+0.46*\sin(24\pi t+\pi/2)$$

=>



Resultierende Wellenform

=>

### Time-varying Orbit:

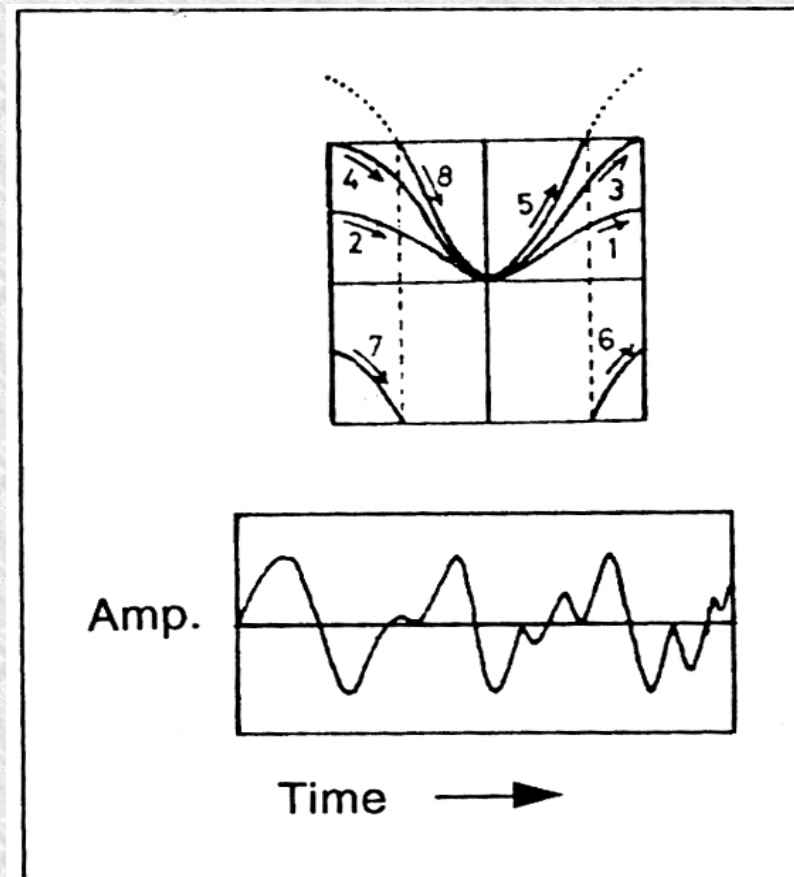
- Durch einen aperiodischen („time-varying“) Orbit und ein fixiertes Wave Terrain, oder durch ein sich änderndes Wave Terrain und einen periodisch-fixierten Orbit, können aperiodische Signale erzeugt werden.

Bsp.: Time-varying Orbit:

nach jedem Durchgang  
wechselt der Orbit  
seine Bahn

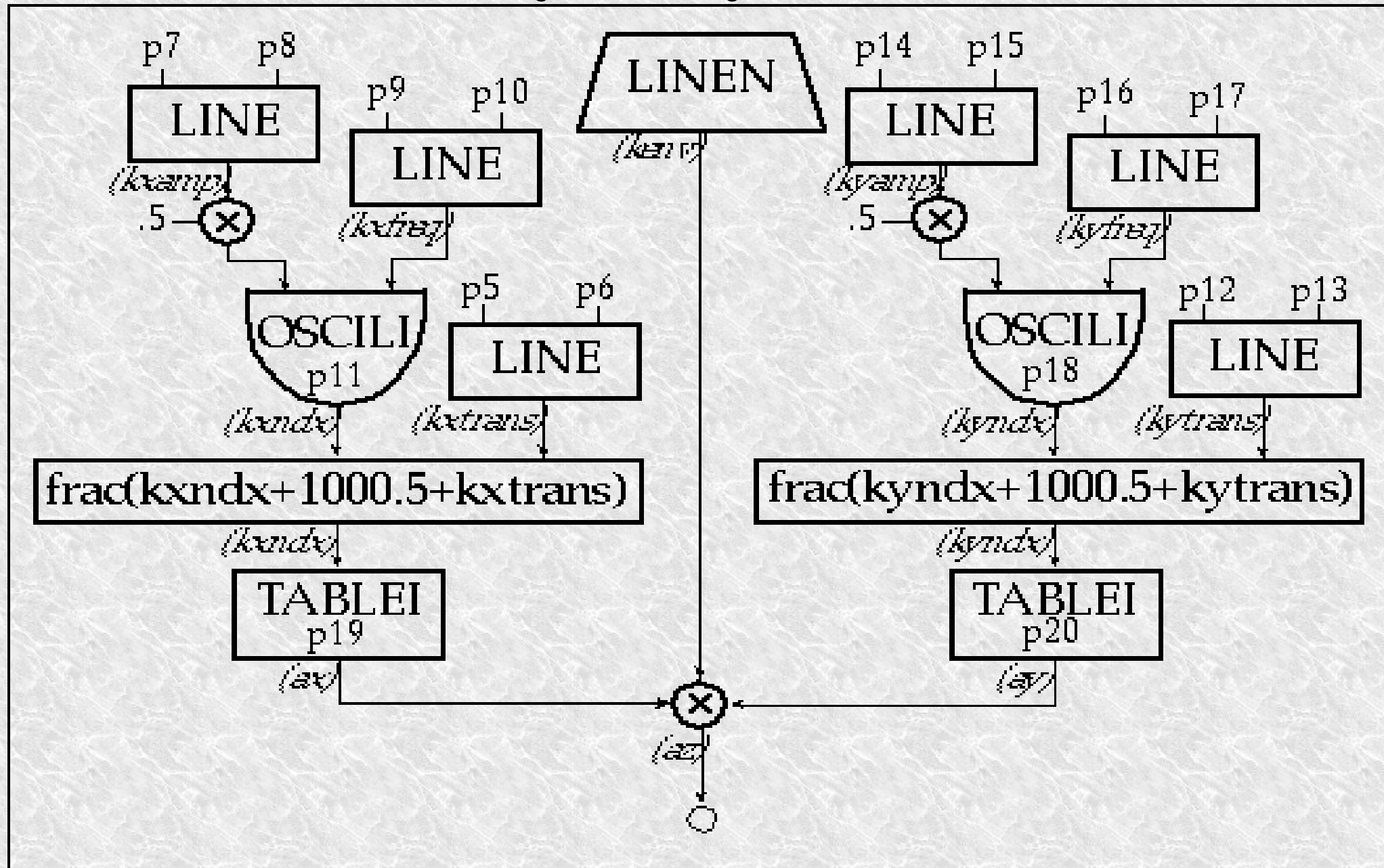
=>

resultierende aperiodische Wellenform =>



# WTS in Csound

Hier das Flussdiagramm zum folgenden Csound-Instrument:



Folgende Parameter sind eingebunden:

```
; p4      =      amp  
; p5-6    =      xtrans: init-final  
; p7-8    =      xoscil amp: init-final  
; p9-10   =      xoscil freq: init-final  
; p11     =      x index f-table  
; p12-13  =      ytrans: init-final  
; p14-15  =      yoscil amp: init-final  
; p16-17  =      yoscil freq: init-final  
; p18     =      y index f-table  
; p19     =      x-axis f-table  
; p20     =      y-axis f-table
```

p5 -p11 = Bewegung des Orbits auf der X-Achse

p12-p18 = Bewegung des Orbits auf der Y-Achse

p19 u. P20 = 2-dimensionale f-tables;

werden im Instrument miteinander multipliziert um die Z-Achse und somit die resultierende Wellenform zu berechnen

Csound-.Orc und -.sco:

WTS.orc

WTS.sco

WTS.aif

Weitere Möglichkeiten WTS in Csound zu implementieren:

<http://csounds.com/jmc/Articles/Wts/WTS.html>

Viel Spaß damit!!