Diplomarbeit

Analoge Programme

Hochschule
Kunsthochschule für Medien, Köln

Student
Andreas Muxel

Fächergruppe
Mediengestaltung

Prüfer
Prof. Dr. Georg Trogemann
Prof. Frans Vogelaar
Martina Höfflin

Köln, Januar 2008

“With the advent of electronics the simple machine, formerly a symbol of progress and modernity, has become more and more an object of nostalgia. The technological revolution in terms of electronics, data-processing and computer techniques (a technology more of brain than muscle) demanded a new artistic model. Up to a point it was supplied by conceptual art, the essential basis of which was the systematic conception and encoding of models and structures. It was based to a certain extent on the systems analysis used in advanced technology. Seen in this light, instead of being a symbol of power and progress, the machine became a metaphor of thought: thought and imagination in terms of mental structures and a production system.”

Irene Veenstra 1985,
in “Machine as a Metaphor for Thinking”

INHALTSVERZEICHNIS

VORWORT

1 MASCHINEN UND IHRE PROGRAMME
1.1 Das rationalistische Denken im mechanistischen Weltbild 011
1.2 Von der klassischen Maschine zur Rechenmaschine 013
1.3 Vermögen zur Abstraktion 016
1.4 Von der Theorie einer Turingmaschine zum heutigen Computer 018
1.5 Mechanisierung von Programmen 021
1.6 Vom Algorithmus zur Interaktion 024

2 DYNAMISCHE UND NICHT-LINEARE SYSTEME
2.1 Der Regelkreis in der Kybernetik 027
2.2 Beschreibung von dynamischen Systemen 029
2.3 Rückkopplung als Steuerungsprinzip 030
2.4 Modell und Simulation 031
2.5 Struktur und Verhalten 032

3 PROGRAMME ALS PROZESSENTWURF
3.1 Programm als Konzept in der Kunst 037
3.2 Formgestaltung vs. Prozessgestaltung 039
3.3 Tatsächliche Bewegung als Ausführung von Programmen 043

4 FAZIT 049

5 VORSTUDIEN
5.1 Ausgangspunkt 053
5.2 Erste Experimente 054

6 CONNECT
6.1 Beschreibung 077
6.2 Aufbau 078
6.3 Conclusio 082

7 ABBILDUNGEN 087

8 LITERATUR 091

VORWORT

Schon während meinem ersten Studium an der Fachhochschule Vorarlberg – Fachbereich InterMedia mit dem Schwerpunkt multimediale Kommunikationsgestaltung – interessierte mich vor allem das Entwerfen von Programmen für rechnergestützte Gestaltungsprozesse. So entwickelte ich bei meiner ersten Diplomarbeit „Formprozessor - Interpretation von Musik und Transformation in Elemente der visuellen Gestaltung“ eine Anwendung zur Visualisierung von akustischen Ereignissen. Wichtig war mir hierbei, dass die Regeln, nach denen die akustische Messung visualisiert wird, jederzeit für den Benutzer sichtbar und somit kontrollierbar waren – sozusagen ein Sichtbarmachen der internen Programmanweisungen auch auf der Benutzeroberfläche. Es war auch interessant zu sehen, dass sehr einfache Regeln in der Übersetzung komplexe und für den Betrachter überraschende Verhalten während dem Ablauf des Programms zeigen können.

Zu Beginn einer neuen künstlerischen Arbeit fällt es mir oft sehr schwer, meine Gedanken zu formulieren, beziehungsweise ist mir nie ganz klar, zu welchem Ergebnis ein Prozess führen wird. Neben dem Experiment mit tatsächlichen Materialien bietet mir auch der Computer eine Möglichkeit, erste Gedanken mithilfe von kleinen selbst geschriebenen Programmen in der Simulation auszuprobieren. Dies hilft mir, meine Gedanken in Worte zu fassen – gezwungener Maßen, da mich ja sonst der Computer nicht verstehen würde.

„Die Kehrseite der Medaille ist der Glaube, man könne nichts programmieren, was man nicht durch und durch verstehe. Dies geht an der Tatsache vorbei, dass das Programmieren, wiederum wie jede Form des Schreibens, in der Mehrzahl der Fälle experimentellen Charakter hat. Man programmiert, genau wie jemand schreibt, nicht, weil man etwas versteht, sondern um zu einem Verständnis zu gelangen. Programmieren ist ein Akt des Entwerfens. Ein Programm zu schreiben bedeutet, einer Welt Gesetze zu geben, die man zunächst in seiner Phantasie erschaffen muss.“ (Weizenbaum 1977, S. 151)

In der Ausführung eines Programmtextes durch den Computer schlägt dann die Schriftform des Programms um in Handlungen und Prozesse. Da diese ersten Schritte in der Programmierung vor allem Experimente sind, lassen sie auch den Freiraum für unbewusste Fehler, die dann wiederum Inspiration für neues

Schaffen sein können. Trotzdem muss der Begriff eines Programms nicht zwingend mit einem Prozess im Computer zusammenhängen. Vielmehr verstehe ich ein Programm auch allgemein als eine Folge von Anweisungen, wie sich Teile eines Systems über die Dauer eines Prozesses verhalten und ihre Zustände aufgrund von bestimmten Aktionen verändern.

Bei rechnergestützten Prozessen taucht immer wieder der Begriff des „Digitalen" auf. In der Literatur finden wir verschiedenste Definitionen, was als „digital" oder „analog" bezeichnet werden kann. Für diese Arbeit beziehe ich mich auf Nake und Grabowski.

„Wir bleiben naiv und setzen das Analoge, wie schon angedeutet, mit dem Kontinuierlichen, das Digitale mit dem Diskreten gleich."
(Nake und Grabowski 2005, S. 139)

Kittler und Goodman treffen sich in dem Punkt, das beide „digital" mit „diskret" und „analog" mit „kontinuierlich" verbinden (vgl. Schröter 2004, S. 28). Deshalb habe ich als Titel dieser Arbeit „Analoge Programme" gewählt, da es mir vor allem um das kontinuierliche Verändern von Zuständen eines Systems über die Zeit und die dafür notwendigen Programme geht.

Ein anderer Grund für die Titelwahl ist auch der, dass ich beim praktischen Teil der Arbeit einfache Programme in tatsächliche Bewegungsabläufe einer kinetischen Skulptur übersetzen will. Der Code in seinem Prozess wird zur Bewegung einzelner Elemente eines realen Objektes. Dadurch, dass die einzelnen Elemente physisch miteinander verbunden werden, soll der Ablauf des Programms nicht nur durch den Rechner, sondern auch durch die Struktur der tatsächlichen Raumskulptur gesteuert werden. Es gibt keine klassische Trennung mehr von Hardware und Software, sondern eine hybride Mischform entsteht.

Wie bereits erwähnt, müssen Programme nicht nur in rechnergestützten Prozessen ihre Anwendung finden. Trotzdem soll im ersten Teil dieser Arbeit ein Rückblick auf die Entstehung der ersten algorithmischen Rechenmaschinen bis zu den heutigen programmierbaren und interaktiven Maschinen erfolgen, da anhand ihrer Entwicklung für diese Arbeit wichtige Begriffe geklärt werden. So soll unter anderem gezeigt werden, dass mit der Entwicklung zur universell-programmierbaren Maschine die Hardware immer weniger wichtig wurde als die Programme, die von ihr ausgeführt wurden. Auch wird versucht deutlich zu machen, dass der ursprüngliche Programmbegriff sehr stark mit algorithmischen Verfahren verbunden war und wir im Laufe der Zeit aber auch eine Entwicklung zu interaktiven Programmen erlebt haben. Wenn wir also einen Algorithmus als Lösungsver-

fahren begreifen, in welchem schon zu Beginn alle Eingabewerte feststehen, so zeichnen sich interaktive Systeme dadurch aus, dass sie zur Laufzeit neue Eingaben verarbeiten können – die Folge können nicht-lineare Programmabläufe sein. Deshalb sollen im zweiten Teil Möglichkeiten der formalen Beschreibung und Regelung solcher Systeme gezeigt werden. Der dritte Teil versucht dann wieder Programme auch als allgemeine Prozessentwürfe zu verstehen und hier werden neben theoretischen Ansätzen auch angewandte Beispiele aus Kunst, Gestaltung und Architektur aufgezeigt. Nach einem abschließenden Fazit dieser ersten drei Teile erfolgt eine Überleitung zum praktischen Teil der Arbeit, welche dann im letzten Kapitel im Detail beschrieben werden soll.

1
MASCHINEN UND IHRE PROGRAMME

1.1 DAS RATIONALISTISCHE DENKEN IM MECHANISTISCHEN WELTBILD

Schon die alten Griechen hatten ein sehr gutes Verständnis dafür, starre geome trische Formen oder Körper im Gleichgewicht formal zu beschreiben. Dennoch hatten sie keine ausreichende Vorstellung von den Gesetzen für die Bewegung von Körpern. Ihnen fehlte eine Theorie für die Weise, in der die Natur tatsächlich die Ortsveränderung der Körper in der Zeit steuert – eine Theorie der Dynamik. Als Galilei 1583 beobachtete, dass ein Pendel als zuverlässiges Zeitmessgerät dienen kann, war dies eine für die Entwicklung der Wissenschaft weitreichende Entdeckung, da nun der Bewegung von Körpern präzise Zeiten zugeordnet werden konnten. Mit der Veröffentlichung der Galileischen „Discorsi“ 1638 trat die neue Wissenschaft der Dynamik in den Vordergrund und der Übergang von antikem Mythenglauben zur modernen Wissenschaft hatte begonnen (vgl. Penrose 1991, S. 157). Die Folge war ein besseres Verständnis dafür, naturwissenschaftliche Phänomene mathematisch zu beschreiben und somit auch berechenbar zu machen.

Dadurch wurde immer mehr ein mechanistisches Weltbild gefördert, indem nur untersucht und als geltend gemacht wurde, was objektiv messbar war. Der erste Schritt dazu bestand in der Abstraktion, was bedeutete, dass nunmehr alle Erfahrungstatsachen unberücksichtigt blieben, die nicht in den jeweiligen Begriffsrahmen passten, innerhalb dessen die Wissenschaft gerade arbeitete. Die Forschung konzentrierte sich ausschließlich auf die die messbaren Aspekte solcher Erfahrungselemente und es entstand ein Gefühl von Macht gegenüber den Naturkräften.

„(...) und als Darstellung der Wirklichkeit ist die naturwissenschaftliche Abbildung der Welt nicht ausreichend, einfach aus dem Grund, weil die Naturwissenschaft nicht einmal den Anspruch erhebt, sich mit Erfahrung schlechthin zu befassen, sondern nur mit bestimmten Ausschnitten und nur in bestimmten Zusammenhängen. (...)

Infolgedessen akzeptieren sie in der Regel das in den naturwissenschaftlichen Theorien implizierte Bild der Welt als vollständige und erschöpfende Darstellung der Wirklichkeit; sie tendieren dazu, diejenigen Aspekte der Erfahrung, die die Naturwissenschaftler wegen mangelnder Kompetenz nicht berücksichtigen, so anzusehen, als seien diese weniger real als jene Aspekte, die die Naturwissenschaft willkürlich durch Abstraktion aus der unendlich reichen Gesamtheit bestehender Tatsachen ausgesondert hat.“
(Huxley 1946, S. 35 f.)

Abstraktion bedeutet deshalb auch immer eine Reduktion des Ganzen – eine Problematik, die wir später nochmal bei der Formalisierung von Programmen finden werden.

Ein wichtiger Schlüssel zum Verständnis dieses rationalistischen Denkens ist das Schaffen von Modellen und Experimenten, mit deren Hilfe zuvor aufgestellte Theorien scheinbar allgemein gültig bewiesen werden können. Der Mensch kann nun aus seiner scheinbaren Machtlosigkeit gegenüber der Natur heraustreten und ein neues aktives Verhältnis zu dieser entwickeln. Formal gesehen ist eine Theorie ein Text, also eine Verkettung von Zeichen eines Alphabets und deshalb in erster Linie eine symbolische Konstruktion. Bereits die Begriffe, mit denen eine Theorie arbeitet, sind Symbole, die wie von Abraham Kaplan treffend beschrieben, nach ihrer Bedeutung in der realen Welt „tasten“ (vgl. Kaplan 1964, S. 296). Er erkennt somit die Problematik, dass die symbolischen Begriffe einer Theorie niemals ihren letzten Grund in der Realität finden können. Der Nutzen einer Theorie besteht nun darin, dass sie die begrifflichen Kategorien vorbereitet, innerhalb deren Fragen formuliert, Modelle von etwas gebildet und eventuell auch spätere Experimente geplant werden können. Der Zweck eines Modells besteht somit nicht darin, die Wirklichkeit in ihrer ganzen Komplexität wieder zu geben, sondern durch eine zuerst rein formale Weise das zu erfassen, was für ein Verständnis eines Aspekts in deren Struktur oder Verhalten bedeutend ist. Somit ist ein Modell letzten Endes ein anderes Objekt als der Gegenstand, den es modelliert. Es gehorcht den Verhaltensgesetzen, die von einer entsprechenden Theorie explizit formuliert sind oder aus ihr abgeleitet werden können (vgl. Weizenbaum 1977, S. 190ff). Das Interessante ist nun auch, dass zwei Modelle im Beobachtungszeitraum dasselbe Verhalten erzeugen, aber auf einer anderen Theorie basieren können. Wenn man nun aber das Verhalten als Beweis einer Theorie sieht, stellt sich natürlich die Frage, welche der beiden Theorien richtig ist, beziehungsweise, ob beide stimmen? Hier kann keine Antwort gegeben werden, vielmehr soll dies nochmal die grundlegende Problematik aufzeigen, dass sich aus dem Verhalten eines Modells nicht immer seine innere Struktur ableiten lässt. Zusammen mit Modellen können nun Experi-

mente geplant werden, deren Aufbau in gewisser Weise aber auch wieder festlegt, in was für einem Rahmen sich das zu Untersuchende bewegen kann.

Eine präzis festgelegte Theorie von dynamischen Gleichungen findet dann im Newtonschen Regelwerk seinen ersten Höhepunkt: Werden die Position, Geschwindigkeit und Massen der verschiedenen Teilchen zu einer Zeit vorgegeben, so sind ihre Positionen und Geschwindigkeiten für alle späteren Zeiten mathematisch determiniert (vgl. Penrose 1991, S. 162). Diese Form des Determinismus ist immer noch kennzeichnend für die klassische Physik und diese rationale Vorgehensweise steht in direkter Tradition zu algorithmischen Problemlösungen und Computerprogrammen. Doch bevor auf diese näher eingegangen wird, soll ein kurzer Rückblick auf die ersten Rechenmaschinen und ihre Entwicklung zu den programmierbaren Maschinen erfolgen.

1.2 VON DER KLASSISCHEN MASCHINE ZUR RECHENMASCHINE

Seit jeher versucht der Mensch Arbeits- und Hilfsmittel zu schaffen, von denen er sich eine Verstärkung der eigenen Kräfte und Fähigkeiten erhofft. Nach Lewis Mumford (vgl. Mumford 1977, S. 219) geht die Erfindung der Maschine auf die Hochkultur des Gottkönigtums in den frühen Perioden des Pyramidenzeitalters zurück.

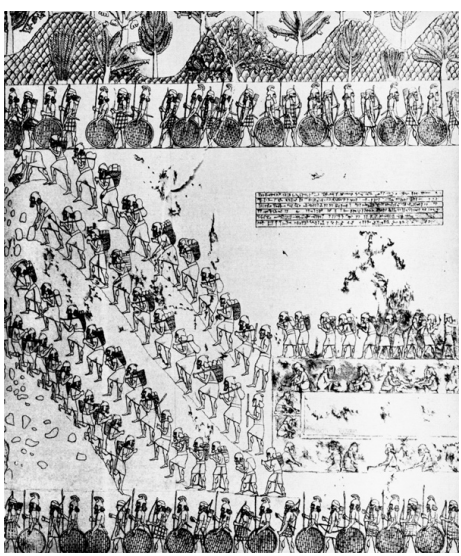


Abb. 1 Archetypische Arbeitsmaschine

Es handelt sich dabei um einen Vorläufer jener mechanischen Maschinen, die wir dann Jahrtausende später wieder in Form von stählernen Ungetümen finden, die das Bild der Industrialisierung geprägt haben. Diese archetypische Arbeitsmaschine besteht in ihren elementaren Bestandteilen aus Menschen und verfügte über eine enorme Produktionskraft, welche wir heute noch zum Beispiel bei der großen Pyramide von Gizeh bewundern können.

„Diese außergewöhnliche Erfindung erwies sich tatsächlich als das früheste Arbeitsmodell für alle späteren komplexen Maschinen, obwohl sich der Schwerpunkt mit der Zeit von den menschlichen Bestandteilen auf die verlässlicheren mechanischen verschob. Die einzigartige Leistung des Königtums bestand darin, das Menschenpotenzial zu konzentrieren und die Organisation zu disziplinieren, was die Ausführung von Arbeiten in nie zuvor da gewesenem Ausmaß ermöglichte.“ (Mumford 1977, S. 219)

Die Befehle der Herrscher stellten sozusagen eine Form von Programm dar, welches dann von Menschenhand mühsam ausgeführt wurde.

Generell können wir sagen, dass mit Maschinen die Einlösung eines erwarteten Handlungs- bzw. Arbeitsvorganges beabsichtigt ist, d.h. die Erfüllung einer spezifischen definierten Aufgabe, durch Umwandlung von Energie in Arbeit. Die physikalische Definition beschreibt die einfache Maschine prinzipiell als System zur Verlagerung von Kraft.

„Maschinen sind physikalische Geräte, die den Angriffspunkt einer Kraft von einer Stelle an eine andere zu verlagern bzw. die Größe oder die Richtung der Kraft oder mehrere Stücke gleichzeitig zu verändern gestattet, um auf diese Weise die Kraft zweckmäßiger einzusetzen.“ (Höfling 1965, S. 145)

Funktionierten die klassischen Maschinen noch nach dem Vorbild des arbeitenden Armes werden dann im 17. Jahrhundert mit der Perfektionierung mechanischer Fertigkeiten auch die ersten Rechenmaschinen konstruiert. Prozesse, die zuvor dem menschlichen Geist vorbehalten waren, konnten nun auch von Maschinen ausgeführt werden. Die älteste bekannte Rechenmaschine stammte von Wilhelm Schickard (1592–1635)

„Dasselbe, was Du rechnerisch gemacht hast, habe ich in letzter Zeit auf mechanischem Wege versucht, und eine aus elf vollständigen und sechs verstümmelten Rädchen bestehende Maschine konstriert,

welche gegebene Zahlen augenblicklich automatisch zusammenrechnet: addiert, subtrahiert, multipliziert und dividiert. Du würdest hell auflachen, wenn Du zuschauen könntest, wie sie Stellen links, wenn es über einen Zehner oder Hunderter weggeht, ganz von selbst erhöht bzw. beim Subtrahieren ihnen etwas wegnimmt.“

Brief von Schickard an Johannes Kepler (von Freytag-Löringhoff 1981, S. 7)

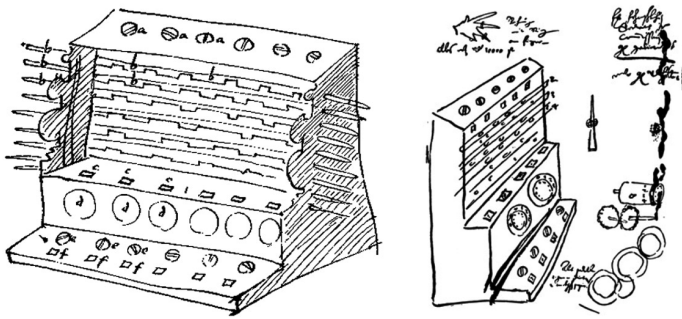


Abb. 2 Skizze der Rechemaschine Schickards

Der Technik wird immer mehr Vertrauen geschenkt – sogar soviel, Prozesse des menschlichen Geistes zu übernehmen. Wenn diese Rechenmaschinen richtig funktionieren, so folgen sie nicht einfach einem arithmetischen Gesetz, sie sind die Verkörperung der arithmetischen Gesetze.

„Wir erwarten zum Beispiel von einem Tischrechner, dass er die Verkörperung der arithmetischen Gesetze darstellt, die wir alle kennen. Sollte er ein in unseren Augen falsches Ergebnis liefern, so ist unser Vertrauen in die Gesetzmäßigkeiten der Maschine so stark, dass wir in der Regel annehmen, wir hätten bei der Eingabe der Daten einen Fehler begangen. Erst wenn sie wiederholt nicht richtig funktioniert, entscheiden wir, dass mit der Maschine irgendetwas nicht stimmt. Es kommt uns keinen Augenblick in den Sinn, dass die Gesetze der Arithmetik etwa aufgehoben sind oder sich geändert haben.“ (Weizenbaum 1977, S. 66)

Aber wie ist es überhaupt möglich, dass ein solch abstrakter Prozess von einer Maschine ausgeführt werden kann?

1.3 VERMÖGEN ZUR ABSTRAKTION

Neben den Fortschritten in der handwerklichen Mechanik war vor allem die zuvor beschriebene rationalistische Tradition und die damit einhergehende Abstraktion und Formalisierung wissenschaftlicher Ergebnisse und technischer Abläufe eine wesentliche Voraussetzung für die heutige programmierbaren Maschine wie den Computer. Dies setzt jedoch die Fähigkeit zur Notation von Zahlen voraus, deren Entwicklung wahrscheinlich schon vor über 30000 Jahren begonnen hat (vgl. Coy 2005, S. 15). Schrift ist als Visualisierung von gesprochener Sprache entstanden und die ersten Zeichen waren Verbildlichungen von Worten und Ideen. Die symbolische Form der Zahlen ist aber wahrscheinlich aus der Erfahrung und als Abstraktion von Ansammlungen gleicher oder fast gleicher Elemente entstanden, wie zum Beispiel der Anzahl der Tiere einer Herde. Die frühen Zählprozesse waren deshalb wohl Zuordnungsprozesse, bevor Aufzählungstechniken entstanden. So wurde die Zahl der Tiere einer Herde durch eine entsprechende Anzahl an Ton-kügelchen in einer Vase festgestellt. Brachte nun ein Hirte eine zu betreuende Herde wieder zurück, so musste deren Größe der Menge der verwahrten Kugeln in der versiegelten Vase entsprechen. Es fand hier also schon eine Abstraktion des Herdentiers zu einem Tonkügelchen statt. Der Prozess des Zählens ließ sich

Abb. 3 Zählen mit Händen und Füßen

vorerst unter Einsatz der Hände und Notfalls auch Füße bewerkstelligen und mit der Zeit gab es auch eine entsprechende Notation – meist mit Hilfe schon vorhandener Schreibnotationen. Im Gegensatz zu Buchstaben, die meist auf Sinneseindrücke verweisen, repräsentierten Zahlen den Prozess des Zählens, also eine rein geistige Tätigkeit (vgl. Coy 2005, S. 17).

Betrachten wir die altgriechische Zahldarstellungen, besitzen diese wie die meisten antiken Zahlennotationen einen gravierenden Mangel: Sie benötigen immer neue Zahlzeichen, um mit dem uneingeschränkten Wachstum, der im Zählprozess angelegt ist Schritt zu halten. Erst eine Stellennotation stellte eine allgemeine Methode zur Erweiterung des Zählraums dar. Hier werden beliebig große Zahlen mit einem kleinen Ziffernvorrat darstellbar. Schon den babylonischen Mathematikern war diese Methode bekannt, allerdings fehlte ihnen ein Zeichen für das Nichts, dass dann auftritt wenn eine Stelle leer bleiben soll. Sie ließen bei Bedarf diese Stelle dann einfach leer besetzt, was aber zu Schwierigkeiten führte, wenn mehrere Leerstellen hintereinander folgten. Ein Existenzrecht unter den Ziffernzeichen erhielt die Null erst viel später, wahrscheinlich zwischen dem 6. und 8. Jahrhundert in Indien. Die Null wird zum Zeichen, um anzuzeigen, dass nichts da ist. Wenn von einer Zahl Null abgezogen wird oder Null abgezählt wird, dann ändert sich nichts. Aber die Null hinter die Zahl gesetzt, verzehnfacht plötzlich ihren Wert. Unser gebräuchliches Zehner-Stellenwertsystem hat ihren Ursprung in den indischen Zahlschriften (vgl. Trogemann und Viehoff 2005, S. 222f).

Zum Zählen genügt ein einzelner Strich, aber ein Stellensystem muss also eine Null und wenigstens ein weiteres Zeichen enthalten. Francis Bacon scheint um 1620 als erster die Frage nach einem minimalen Code untersucht zu haben und entdeckte, dass alle Lettern des Alphabets mit nur zwei Zeichen, also einem Binäralphabet beschreibbar sind. Er nutzte diese Erkenntnis für die Verschlüsselung von Nachrichten, erkannte aber noch keinen weiteren Nutzen für seine Entdeckung. Erst Gottfried W. Leibniz begriff um 1697, dass somit alles, was bisher mit dem geläufigen Alphabet und dem Dezimalsystem notierbar war, nun auch binär konstruiert werden kann. Denn die Operationen der Addition und Multiplikation lassen sie wie ihre Umkehroperationen der Subtraktion und der Division auch im binären Zahlkörper vollziehen. Er erkennt, dass diese binäre Arithmetik eine einfache maschinelle Verarbeitung zuließe (vgl. Coy 2005, S. 19). Dennoch sollte es noch einige Zeit dauern, bis dieses Konzept von binären Rechenmaschinen tatsächlich umgesetzt wurde.

1.4 VON DER THEORIE EINER TURINGMASCHINE ZUM HEUTIGEN COMPUTER

Die ersten Rechenmaschinen ab dem 17. Jahrhundert waren nicht programmierbar, da bei ihrer Konstruktion kein eigener Bereich für die Spezifizierung von Operationen vorgesehen war – vielmehr ist der Mechanismus für die Ausführung der Operationen die Konstruktion der Maschine selbst. Wir können erst von einer „Programmierbarkeit" der Maschine sprechen, wenn ihr konkretes Verhalten geändert werden kann, ohne nochmals etwas an ihrer Konstruktion zu verändern. Die Grundidee der Programmierung besteht darin, dass Zeichen zu Befehlen werden, die dann wieder von einer Maschine ausgeführt werden können. Der erste Schritt dazu ist die Abstraktion, wie wir sie schon in den Naturwissenschaften des mechanistischen Weltbildes kennen gelernt haben (vgl. S. 11f). Unter dem nächsten Schritt, der Formalisierung, versteht man die Übersetzung der Abstraktion in eine universelle Sprache, deren Zeichen ihre eigene Realität bilden, in der sie nur noch sich selbst repräsentieren (vgl. S. 17). Mit ihr können dann genaue Anweisungen zur Lösung eines Problems beschrieben werden. Dieser als „effektives Verfahren" oder „Algorithmus" bezeichneter Begriff ist einer der wichtigsten in der modernen Mathematik.

Der deutsche Mathematiker Davis Hilbert forderte im Jahre 1928 auf dem internationalen Kongress in Bologna ein allgemein algorithmisches Verfahren für die Lösung von mathematischen Fragen – oder besser gesagt für eine Antwort auf die Frage, ob ein solches Verfahren im Prinzip überhaupt existiert. Was nun den englischen Mathematiker Alan Turing an dem Hilbertschen Problem interessierte, war die Frage, ob es ein allgemein mechanisches Verfahren gibt, das im Prinzip alle Probleme der Mathematik – auch als „Entscheidungsprobleme" bezeichnet – eines nach dem anderen zu lösen vermag (vgl. Penrose 1991, S. 32). Turing entwickelte erstmals das Konzept für eine Rechenmaschine, die alles berechnen, was überhaupt berechenbar ist, ohne dass für jede neue Aufgabe in die innere Struktur der Maschine eingegriffen werden muss. Seine universelle Rechenmaschine kann jede andere Rechenmaschine simulieren und ist somit universell einsetzbar. Bevor auf diese universelle Turingmaschine eingegangen wird, soll allerdings noch der wesentliche Aufbau einer speziellen Turingmaschine gezeigt werden. Im wesentlichen besteht diese aus drei Einheiten (vgl. Trogemann und Viehoff 2005, S. 244f):

1. Einem einseitigen Band, das in Zellen unterteilt ist, wobei jede einzelne Zelle genau ein Zeichen aus einem bekannten Zeichenvorrat annehmen kann. Da die Anzahl der Eingaben theoretisch unbegrenzt ist, müsste auch das Band unendlich lang sein.

2. Einem beweglichen und programmgesteuerten Schreib-Lesekopf, der sich zu einem Zeitpunkt nur über einem Feld befinden kann, welches er lesen oder beschreiben kann.
3. Einer Kontrolleinheit, deren Merkmal „Zustände“ sind, in denen sich der Mechanismus befinden kann und zwischen denen er wechselt.

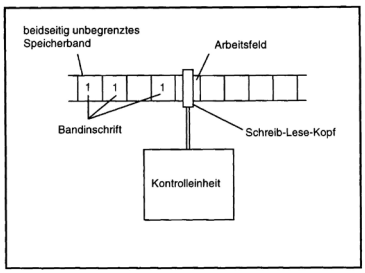


Abb. 4 Schema einer speziellen Turingmaschine

Das Verhalten, also die Folge von Zuständen, die angenommen werden können und die Zeichen, die geschrieben werden, hängt vom momentanen Zustand der Maschine ab und dem Zeichen, das gerade auf dem Band betrachtet wird. Davon abhängig wird die Maschine ein neues Zeichen auf das Band schreiben, in einen anderen Zustand übergehen und sich auf dem Band um ein Feld nach links oder rechts bewegen. Durch dieses einfache Verhalten lassen sich verschiedene arithmetische Operationen jeweils durch spezielle Turingmaschinen durchführen. So können bestimmte Zustände einer Maschine als Endzustände definiert werden und sobald sie in einen dieser Zustände kommt, bleibt sie stehen und liefert das Ergebnis auf die Eingabe. In seinem Paper „On Computable Numbers, with an Applicaton to the Entscheidungsproblem“ (1937) entwirft Turing schließlich das Prinzip der Gleichbehandlung von Programmanweisungen (Operatoren) und Daten (Operanden) als Konzept einer universell-programmierbaren Maschine. Indem man eine universelle Turingmaschine zuerst mit einem entsprechenden Programm versorgt, welches im Binärsystem auf das Band geschrieben wird, kann sie dazu gebracht werden, das Verhalten jeder beliebigen speziellen Turingmaschine zu imitieren (vgl. Penrose 1991, S. 48ff). Diese Universalität zeichnet die Mächtigkeit des Konzeptes von Turing aus, allerdings sei nochmal darauf hingewiesen, dass eine Turingmaschine kein physikalisches Objekt, sondern ein Stück „abstrakte Mathematik“ ist.

Es war dann vor allem John von Neumann, der das Prinzip Turings der Gleichbehandlung von Daten und Programm und das damit verbundene binäre Konzept aufnahm und in seiner Von-Neumann-Architektur umsetzte, die immer noch Vorbild für die heutigen elektronischen Rechner ist. So beschreibt er 1945 in „First Draft of Report on the EDVAC“ einen ersten Entwurf für eine speicherprogrammierte Maschine. Durch Gleichbehandlung von Operatoren und Operanden in einem gemeinsamen Speicher ist es in diesen Maschinen erst möglich, dass sich ein Programm selber lesen und modifizieren kann.

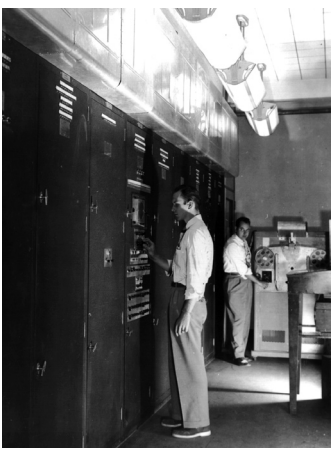


Abb. 5 EDVAC

Aber kann nun ein solches binäre System universell alles in der realen Welt Vorkommende simulieren?

„Man kann jedoch mit Fug und Recht die Universalität des digitalen Codes in Frage stellen: Einerseits weil sie nur bis zur Grenze dessen reicht, was sich überhaupt formalisieren und berechnen lässt. (...) Anderseits bestimmen Architektur und Geschwindigkeit digitaler Hardware, welche Formalisierungen ausgeführt werden können. So hat die Von-Neumann-Architektur spezifische Limitationen: Zwischen dem Speicher und dem Central Processing Unit gibt es in der Regel nur einen Datenbus, d.h. Programme werden strikt sequentiell ausgeführt (so genannter ‚Von-Neumann-Flaschenhals‘). Diese Sequenzialität verlangsamt z.B. bestimmte Formen der Verarbeitung höherdimensionaler Daten: Das wären z.B. Bilder...“ (Schröter 2004, S.13)

Eine Alternative zur sequentiellen Abarbeitung von Programmanweisungen durch einen Zentralprozessor stellt die Idee des Parallelrechners dar, die auch hier kurz der Vollständigkeit halber erwähnt werden soll. So entwickelte unter anderem schon Heinz von Foerster ab 1958 am Biological Computer Lab erste Ansätze für parallele Rechner und orientierte sich hier stark am Nervensystem von Lebewesen.

„Ein paralleler Rechner ist zum Beispiel das Auge. Du hast zehn Millionen Zäpfchen und Stäbchen auf der Retina, die alle gleichzeitig beleuchtet werden und dann ihre Signale weiterschicken, wo alles parallel gerechnet wird und parallel ins Hirn läuft. (...) Wenn du parallel rechnest, kann der einzelne Rechner sehr langsam sein, wie zum Beispiel ein Neuron. Ein Neuron ist ja nicht so schnell wie ein Mikrochip in einem Rechner. Ein Mikrochip kann in einer millionstel Sekunde rechnen; ein Neuron braucht ungefähr eine zehntel Sekunde, bevor es ‚Mu' sagen kann. Aber wenn du mit zehn Millionen Neuronen gleichzeitig rechnest, bist du hundertmal schneller als ein Mikrochip. Also wenn du parallel rechnest, kannst du sehr viel mehr in kürzerer Zeit ausrechnen." (von Foerster 2002, S. 64)

Solche Parallelrechner werden deshalb unter anderem in der Erkennung von Muster, wie zum Beispiel der rechnergestützten Spracherkennung, Texterkennung und Gesichtserkennung verwendet.

1.5 MECHANISIERUNG VON PROGRAMMEN

Nachdem wir im vorherigen Kapitel unter anderem die Grundprinzipien für die Abstraktion und Formalisierung von Programmen und deren algorithmische Ausführung gesehen haben, soll nun nochmal genauer auf die eigentliche Mechanisierung von Programmen, d.h. deren tatsächliche Ausführung auf der Maschine eingegangen werden.

So finden wir mit Charles Babbages „Analytical Engine" (1837) schon 100 Jahre vor Turings Konzept einer universell-programmierbaren Maschine einen ersten Entwurf für eine code-gesteuerte Maschine, die algorithmische Verhalten zu realisieren vermochte. Als maschinenlesbarer Datenträger für die Programmanweisungen sollten Lochkarten verwendet werden und es kam zu einer Ablösung des Programms von der Hardware. Bereits 30 Jahre zuvor hatte J.M. Jaquard bei seinen um 1800 erfundenen Webstühlen Lochstreifen eingesetzt, die das Weben unterschiedlicher Muster automatisch steuerten.

„Die Analytical Engine webt algebraische Muster, so wie Jaquards Webstuhl Blumen und Blätter webt.“
(Countess of Lovelace, zitiert nach Dotzler 1996)

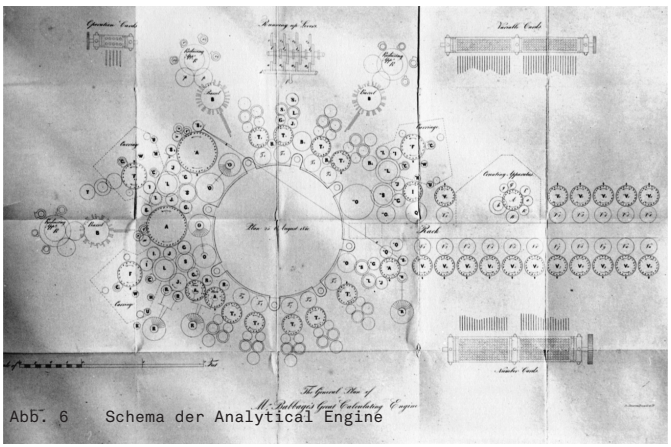


Abb. 6 Schema der Analytical Engine

Verstehen wir unter einem Programmcode die Abbildung einer endlichen Menge von Befehlszeichen, welche die Zustandsänderungen der Maschine steuern, dann wird klar, dass seine Maschine unserem heutigen Begriff von Programmierbarkeit sehr nahe kommt. Wie im vorherigen Kapitel erwähnt (vgl. S. 19), ermöglicht vor allem die Trennung der Software von der Hardware eine Programmierung dieser.

Im Hinblick auf die Grundprinzipien algorithmischer Maschinen gab es bei der Ablösung des mechanischen Zeitalters durch das elektronische so gut wie keine Weiterentwicklung. Die gleichen algorithmischen Denkmuster manifestieren sich lediglich in einem neuen Material. Beim Prozessieren eines Programms durch die Maschine wird immer ein Aktivitätsträger gebraucht, der die Handlungsanweisungen tatsächlich ausführt. Ob dieser Aktivitätsträger nun elektronisch, mechanisch oder organisch ist, bleibt zweitrangig. Dies zeigt Turings Konzept der universell-programmierbaren Maschine, da er hier theoretisch beweisen konnte, dass jede Maschine, deren Hardware ein bestimmtes Minimum an Kompliziertheit und Flexibilität erreicht, jede andere Maschine durch spezielle Programmanweisungen simulieren kann (vgl. Penrose 1991, S. 21f).

„Es ist möglich, den Effekt einer Rechenmaschine zu erreichen, indem man eine Liste von Handlungsweisen niederschreibt und einen Menschen bittet, sie auszuführen. Eine derartige Kombination eines

Menschen mit geschriebenen Instruktionen wird ‚Papiermaschine' genannt. Ein Mensch, ausgestattet mit Papier, Bleistift und Radiergummi sowie strikter Disziplin unterworfen ist in der Tat eine Universalmaschine." (Turing, zitiert nach Tarnas 1997, S. 529)

Somit wird die Hardware immer weniger wichtig als die Software, also die Programme oder Algorithmen, die auf der Maschine prozessiert werden. So sind nur noch die Ein-/Ausgaberelationen und die internen Zustandsübergänge einer Maschine entscheidend (vgl. Trogemann und Viehoff 2005, S. 82). Die Maschine als solche verliert immer mehr ihren eigentümlichen Charakter zu Gunsten ihrer universellen Funktionalität. Ein Punkt auf den später nochmal bei der Beschreibung des praktischen Teils dieser Arbeit eingegangen werden soll.

Die Berechnung von diskreten Algorithmen setzen Maschinen voraus, die nur eine endliche Anzahl von Stellungen einnehmen können. Diese möglichen Stellungen heißen im Allgemeinen „Zustände" einer Maschine. Bei mechanischen Rechenmaschinen gibt es immer unkontrollierte Schwankungen im System, in der Kommunikationstheorie auch als Rauschen bekannt, da hier ein solcher Zustand nie einer endlich Anzahl von Stellungen entsprechen kann. Beim elektronischen Rechner ist dies aber der Fall. Der Prozessor kann hier die Maschine in diskrete Zustände – auch als „digital" (vgl. Nake und Grabowski 2005, S. 142) bezeichnet – bringen, wobei der physikalische Übergang zwischen zwei Zuständen immer noch kontinuierlich – auch als „analog" bezeichnet (vgl. ebenda, S. 142) – passiert. Somit ist ein elektronischer, digitaler Rechner zwar in seinem Ablauf auch immer noch an die physikalischen Gesetze der realen Welt gebunden, seine Genauigkeit ist aber bei den Zuständen kontrollierbar. So kann diese jederzeit durch Hinzufügen weiterer Stellen erhöht werden. Soll allerdings von einem digitalen Rechner Zufall und Chaos erzeugt werden, stoßt sein diskretes Funktionsprinzip hier wieder an seine Grenzen, da die Maschinen immer mit einer endlichen Anzahl von Zuständen operiert. Hier bieten die unkontrollierbaren Schwankungen von analogen, mechanischen Systemen wieder große Vorteil. Dies ist ein wichtiger Aspekt, der im praktischen Teil der Arbeit wieder aufgegriffen werden soll. Zudem wurde die beim digitalen Rechner eingesetzte Technologie über die Zeit immer kompakter und schneller und wir nehmen in der Ausführung der Programme die physikalischen Übergänge von einem Zustand in einen anderen direkt gar nicht mehr wahr. Die eigentliche Maschine verliert zunehmend ihren eigenen Charakter und wird immer mehr zur anonymen Black Box für den äußeren Betrachter.

Es muss auch bedacht werden, dass Rechenmaschinen ausschließlich Berechnungen über einen bestimmten formalisierbaren Bereich ihrer realen Umwelt anstellen. Will man zum Beispiel Messungen in der realen Welt als Eingabewerte

einem Programm zuführen, geschieht dies über einen Analog/Digital-Wandler, der die kontinuierlichen Werte in diskrete für den Rechner interpretierbare Zeichen umwandelt. Da diese nun abstrahierte Außenwelt im System nur noch in Form von Daten repräsentiert wird, führt diese Berechnung natürlich nicht direkt zu einer Zustandsänderung in der Außenwelt. Soll durch die Berechnung aber wieder eine tatsächliche Handlung ausgeführt werden, zum Beispiel das Vor- oder Zurückfahren eine Elektromotors, brauchen wir wieder einen Digital/Analog-Wandler, der dies möglich macht. Ein Punkt der bei der Umsetzung der praktischen Arbeit von großer Bedeutung ist.

1.6 VOM ALGORITHMUS ZUR INTERAKTION

Wenn wir nochmals das Newtonsche Regelwerk zur Mechanisierung der Welt betrachten (vgl. S. 13), dann lag sein Erfolg vor allem darin, dass bis auf die Feststellung der Anfangsbedingungen die Abläufe mathematisch determiniert waren. Die Sprache, durch die der Determinismus seinen mathematischen Ausdruck findet, sind die Differentialgleichungen. Obwohl man heute weiß, dass die Newtonsche Mechanik aufgrund ihrer starken Sensibilität gegenüber den Anfangsbedingungen keine genaue Vorhersagbarkeit des Verhaltens ihrer Modelle liefern kann – ein Punkt auf den später nochmal genauer eingegangen werden soll – sind die Differentialgleichungssystem dennoch bis heute die mächtigsten Werkzeuge der Naturwissenschaften. Dieser Determinismus steht wie bereits erwähnt in direkter Tradition zu den algorithmischen Problemlösungen und ihrer Ausführung als Computerprogramme (vgl. S. 18).

So stehen Befehle und Daten schon zu Beginn der Berechnung fest und werden im Speicher des Rechners eingeschrieben. Der Begriff der „Zeit“ wird in der klassischen Algorithmentheorie nur durch die Anzahl der notwendigen Berechnungsschritte festgelegt. Der große Vorteil des digitalen Rechners ist nun unter anderem, dass er die einzelnen Schritte eines Algorithmus sehr schnell ausführen kann, was in der Praxis große Vorteile mit sich bringt, zum Beispiel bei rechnergestützten Simulationen. Bei einem algorithmischen Verfahren sind es allerdings nur klar spezifizierbare Aufgaben, die durch eine endliche Folge von Handlungsanweisungen gelöst werden können. Die Handlungsanweisungen stehen vor Beginn der Berechnung fest und werden im Verlauf der Berechnung auch nicht mehr verändert. So bezeichnet der Physiker Heinz Pagel solche einfache dynamische und lineare Systeme auch als „simulierbar“, da sie mathematisch exakt beschrieben werden können (vgl. Frazer 1995, S. 107).

Somit gibt der Formalismus hier bereits den Rahmen vor, innerhalb dessen sich die Bearbeitung von Eingabegrößen bewegt (vgl. S. 11f) – nach der Ausführung eines Programmschrittes ist klar definiert, welcher Schritt als nächstes ausgeführt wird. Durch ein solches Programm können deshalb nur Fragen beantwortet werden, die klar formulierbar sind und somit frei von jeglicher Interpretation sind.

„Aber durch die Wahl des formalen Rahmens sind die Fragen bereits entschieden, egal, ob wir die einzelnen Folgen, die sich aus der Wahl des Rahmens ergeben, schon sehen können oder ob wir sie erst langwierig von Hand oder maschinell ableiten müssen.“ (Trogemann und Viehoff 2005, S. 71f).

Wie eben gezeigt ist beim Ausführen eines Algorithmus keine Unterbrechung des Programms vorgesehen. In der Geschichte des Computers fand aber ein Paradigmenwechsel vom Algorithmus zur Interaktion statt.

„The evolution of computer technology from the 1970s to the 1990s is captured by a paradigm shift from algorithms to interaction. Algorithms yield outputs completely determined by their inputs that are memoryless and history independent, while interactive systems like personal computers, airline reservations systems, and robots provide history-dependent services over time that can learn from ad adapt from experience.“ (Wegner 1996)

Eine Programm ist also dann interaktiv, wenn es die Fähigkeit besitzt, zur Laufzeit neue Eingaben zu verarbeiten. Wie können aber solche nicht-lineare und dynamische Systeme beschrieben und geregelt werden? Darauf soll im nächsten Kapitel näher eingegangen werden.

2
DYNAMISCHE UND NICHT-LINEARE SYSTEME

2.1 DER REGELKREIS IN DER KYBERNETIK

Obwohl die Ursprünge der Kybernetik unterschiedlich tief in die Geschichte zurückreichen, können die so genannten „Macy-Konferenzen", benannt nach ihrem Förderer, der Josiah Macy Jr. Foundation als Beginn der Kybernetik angesehen werden. So stand schon bei der ersten Konferenz 1946 genau fest, was nach einer gemeinsamen Theorie verlangte: die Prinzipien der aktuellen Computergeneration, die jüngsten Entwicklungen der Neurophysiologie und ein auf den Menschen bezogenes Feld von Psychiatrie, Anthropologie und Soziologie. Wissenschaftler aus den verschiedensten Disziplinen wurden eingeladen um zuvor Spezielles zusammenzudenken und übergreifende Ordnungen zu entwerfen" (vgl. Pias 2004, S. 10).

"Ich glaube, die Kybernetik ist der größte Bissen aus der Frucht vom Baum der Erkenntnis, den die Menschheit in den letzten zweitausend Jahren zu sich genommen hat." (Bateson 1985, S. 612)

Die Kybernetik muss deshalb als eine Wissenschaft von den Wissenschaften gesehen werden. Zwischen 1946 und 1948 noch unter dem etwas umständlichen Titel „Circular Causal, And Feedback Mechanisms in Biological and Social Systems" abgehalten, heißen die Konferenzen ab 1949 programmatisch „Cybernetics", inspiriert nach dem Titel des Buches „Kybernetik – Regelung und Nachrichtenübertragung im Lebewesen und in der Maschine", dass einer der Teilnehmer, Norbert Wiener, kurz zuvor veröffentlicht hatte.

In der Kybernetik – mit ihrem Definition abgeleitet vom griechischen Wort „Kybernetes" für Steuermann – stehen nicht mehr deterministische und lineare Systeme im Fokus der Betrachtung, sondern Systeme, bei denen ein Teil des Ausgangs in das System zurückgemeldet wird, um damit auf das Verhalten des Systems einzuwirken und insbesondere Gleichgewichtszustände aufrecht zu erhalten oder wieder herzustellen. Eines der zentralen Prinzipien der Kybernetik ist die Zirkularität. So werden ausschließlich organisatorisch geschlossene Systeme behandelt,

die auf der Basis rückgekoppelter Kausalität ihr eigenes Verhalten regeln. Somit hängt die Zukunft eines Prozesses nicht nur von externen Systemeingaben ab, sondern in gleichem Maße auch von der eigenen Vergangenheit des Systems.

Die Kybernetik kann die Frage, was etwas ist, von dem Tag an beantworten, an welchem sie ein Modell davon bauen kann (vgl. Moles 1959, S. 8). Wichtig für das kybernetische Modell ist der Regelkreis (englisch: „feedback system"). Ein Regelkreis zeichnet sich dadurch aus, dass das Ergebnis eines Vorgangs die Ausgangsbedingungen beeinflusst und wieder steuert. Durch ständige Kontrolle des Ist-Zustandes und entsprechende Regulierung soll eine Annäherung an den Soll-Zustand des Systems erfolgen. Ein in der Literatur immer wieder gerne genanntes Beispiel ist der Thermostat, welcher die Temperatur in einem Raum misst, mit der gewünschten Temperatur vergleicht und die entsprechenden Maßnahmen einleitet, um die Ausgangsbedingungen anzupassen. Der Thermostat ist sozusagen das System, welches versucht eine bestimmte Ordnung (hier eine konstante Temperatur) aufrecht zu erhalten. Dazu vergleicht er permanent die Umwelt mit dem in ihm angelegten Wert und versucht die Umwelt so zu gestalten, dass sie diesen Wert annimmt, zum Beispiel durch Senken oder Steigern der Wärmezufuhr.

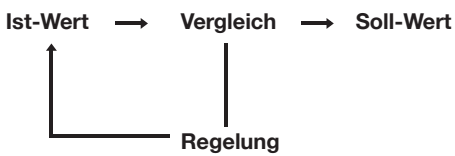


Abb. 7 Regelkreis in der Kybernetik

Die Kybernetik ist auch eine Wissenschaft der Organismen und ihrer Wirkung auf die Umwelt. Allerdings lässt sie bewusst offen, ob der zu untersuchende Mechanismus aus lebenden Zellen besteht, aus einer Gesamtheit chemischer Reaktionen, aus einer Gruppe kollektiv handelnder Individuen, aus mechanischen Verzahnungen oder elektronischen Relais (vgl. ebenda, S. 8). Sie ist vor allem eine Wissenschaft von Modellen und dadurch dass hier die Materialität in erster Hinsicht nicht wichtig ist, ist das Modell des Regelkreises auf alle möglichen Bereiche anwendbar. Dieser Generalismus ist aber auch einer der späteren Kritikpunkte an der Kybernetik und sie löst sich im Laufe der Zeit wieder in die verschiedensten Disziplinen auf, unter anderem in die „Allgemeine Systemtheorie", die sich dann später als eigener Bereich in den „Systemwissenschaften" weiterentwickelte, auf die nun näher eingegangen werden soll, weil sie sich sehr gut eignet, dynamische Systeme zu beschreiben.

2.2 BESCHREIBUNG VON DYNAMISCHEN SYSTEMEN

Was genau als System bezeichnet wird, ist nicht allgemein gültig geklärt und es gibt die verschiedensten Definitionen, abhängig aus welchem Bereich der Wissenschaft die Definition kommt (vgl. Matthies 2002, S. 2). Generell beschreibt ein System ein Gebilde, das aus vielen verschiedenen Teilen aufgebaut ist, zwischen denen Beziehungen bestehen. Die Systemwissenschaft untersucht nun die Zusammenhänge und Wechselwirkungen in zeitsensitiven, also dynamischen Systemen, um zu einem besseren Verständnis des Verhaltens zu gelangen. Das Verhalten eines Systems ist seine Reaktion auf innere und äußere Veränderungen und meist immer direkt beobachtbar, wobei die eigentliche Struktur eines Systems beim äußeren Blick meist verborgen bleibt.

Es ist wichtig zu berücksichtigen, dass das, was als System anzusehen ist, immer nur vom Beobachter entschieden wird und keine innewohnende Eigenschaft des Systems ist. Somit gibt es auch Dinge, dich sich außerhalb des Systems befinden. Hier spricht man von der Systemumgebung, in die ein System dann eingebettet ist. Als Systemgrenze wird der Bereich bezeichnet, indem sich Systemumgebung und System berühren. Diese Trennung zwischen System und Systemumgebung ist meist nie eindeutig festzulegen, gerade wenn es sich um ein offenes System handelt. In diesem Fall nimmt das System Materie oder Energie plus Materie von seiner Umwelt auf oder gibt dies an seine Umgebung ab (vgl. ebenda, S. 10).

Es gibt einen Unterschied zwischen dem direkt beobachtbaren Systemverhalten und seinem inneren Zustand. Die Beobachtung des Systemverhaltens reicht deshalb nicht aus, um das zukünftige Verhalten des Systems vorhersagen zu können. Meist erst mit der Kenntnis des Systemzustandes lassen sich sinnvolle Schlussfolgerungen bezüglich des zukünftigen Verhaltens ziehen. Hierzu werden die Zustandsgrößen benötigt, welche die essentiellen, den Systemzustand zu jedem Zeitpunkt vollkommen beschreibenden Größen darstellen. Die Zustandsgröße stellt somit die Summe der Zustandsänderungen über den betrachteten Zeitraum dar – sie ist sozusagen die messbare Geschichte des Systems. Zustandsänderungen können aufgrund äußerer Einwirkung oder von Prozessen innerhalb des Systems zustande kommen (vgl. ebenda, S. 11). Wirken sich Zustandsgrößen wieder auf die Zustandsveränderungen des Systems aus, so spricht man von Rückkopplung. Da dies ein wichtiger Bestandteil in der praktischen Arbeit sein wird, soll hierauf nochmal näher eingegangen werden.

2.3 RÜCKKOPPLUNG ALS STEUERUNGSPRINZIP

Wie wir in der Kybernetik schon gesehen haben (vgl. S. 27f), kann über einen Rückkopplungskreis die Dynamik und Stabilität eines Systems gesteuert werden. Rückkopplungen erhöhen die Komplexität des Systemverhaltens und es lässt sich die Entwicklung und das Verhalten sehr schwierig vorhersagen, gerade wenn es sich um ein offenes System handelt (vgl. S. 29). Über eine Rückkopplung läuft eine am Anfangspunkt aufgegebene Störung wieder ins System zurück. Es wird zwischen positiver und negativer Rückkopplung unterschieden.

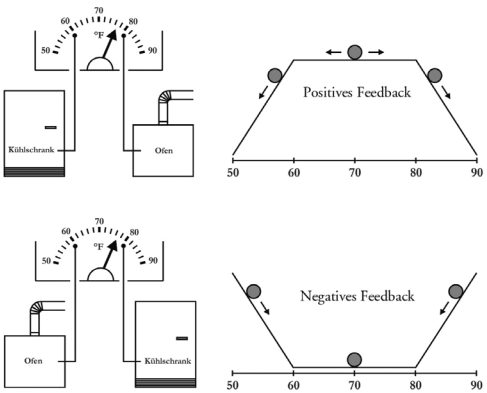


Abb. 8 Positives und Negatives Feedback

Eine positive Rückkopplung, auch Mitkopplung genannt, erfolgt dann, wenn eine steigende oder fallende Zustandsgröße wieder in das System zurückgeführt wird, und somit zu einer weiteren Steigerung beziehungsweise einem Abfall dieses Wertes führt. Daher bedingt eine Mitkopplung eine Selbstverstärkung des Systems und nach einer gewissen Zeit kann sie auch zum kompletten Systemzusammenbruch, also der Katastrophe führen.

Eine negative Rückkopplung, auch Gegenkopplung, erfolgt hingegen dann, wenn eine hohe oder niedrige Zustandsgröße und seine Rückführung ins System dazu führen, dass sich der neue Wert der Zustandsgröße verringert beziehungsweise erhöht. Deshalb führt eine Gegenkopplung zu einem stabilen und gegenüber Störungen aus der Umwelt widerstandsfähigen Systemverhalten. In einem dynamischen System oszilliert deshalb die aktuelle Zustandsgröße bei einer negativen Änderungsrückkopplung um ihren Anfangswert. Negative Rückkopplungsmecha-

nismen scheinen weit verbreitet zu sein, in ökologischen Systemen genauso wie in ökonomischen, in biochemischen Modellen genauso wie in Modellen der Psychologie. Und auch in der Steuer- und Regelungstechnik findet dieses Prinzip der Dämpfung ihre Anwendung. Beim praktischen Teil dieser Arbeit sollen positive und negative Rückkopplungsmechanismen genutzt werden um die Bewegungen von bestimmten Elementen des Systems anzuregen, beziehungsweise zu beruhigen und stabilisieren.

2.4 MODELL UND SIMULATION

Da während der Umsetzung der praktischen Arbeit auch die rechnergestützte Simulation Teil des Gestaltungsprozesses war, soll hier nochmals allgemein auf Modellbildung und Simulation eingegangen werden. Damit ein relevanter Ausschnitt der Realität näher untersucht werden kann, benötigen wir in der Regel eine vereinfachte Abbildung, auch als Modellbildung bezeichnet (vgl. S. 11f). So beschreibt Herbert Strachowiak drei Hauptmerkmale des allgemeinen Modellbegriffs (vgl. Stachowiak 1973, S. S. 131f):

1. Abbildungsmerkmal: Ein Modell ist immer ein Abbild von etwas, eine Repräsentation natürlicher oder künstlicher Originale, die selbst wieder Modelle sein können.
2. Verkürzungsmerkmal: Ein Modell erfasst nicht alle Attribute des Originals, sondern nur diejenigen, die dem Modellschaffer bzw. Modellnutzer srelevant erscheinen.
3. Pragmatisches Merkmal: Pragmatismus bedeutet soviel wie Orientierung am Nützlichen. Ein Modell ist einem Original nicht von sich aus zugeordnet. Die Zuordnung wird durch die Fragen Für wen?, Warum? und Wozu? relativiert. Ein Modell wird vom Modellschaffer bzw. Modellnutzer innerhalb einer bestimmten Zeitspanne und zu einem bestimmten Zweck für ein Original eingesetzt. Das Modell wird somit interpretiert.

Systemmodelle bilden die Wirkungsstruktur so ab, dass das Verhalten eines Systems erklärt werden kann. Die Nachbildung eines dynamischen Systems über Modelle wird als Simulation bezeichnet. Ziel von Systemanalyse, Modellbildung und Simulation ist es, eine möglichst zuverlässige, stellvertretende Verhaltensbeschreibung eines realen Systems zu erhalten. Wird eine Simulation von einem Computer ausgeführt, so wird die Simulation zum Programm, d. h. eine Einheit von Handlungsanweisungen. In der Simulation hat die virtuelle die analoge (mechanische, elektrische, hydraulische) Simulation beinahe vollständig abgelöst (vgl. Bossel 1994, S. 13). Hierfür gibt es eine ganze Reihe von Gründen.

Dadurch, dass alles zu Betrachtende bei der Abstraktion und Formalisierung in eine Sprache übersetzt wird, deren Zeichen ihre eigene Realität bilden, in der sie nur noch sich selbst repräsentieren (vgl. S. 18), kann eine einheitliche Methodik eingesetzt werden, die völlig von der Art des betrachteten Systems getrennt ist. Diese Programme sind somit universell einsetzbar. Auch sind die Kosten der digitalen Modellerstellung und Simulation im allgemeinen nur ein Bruchteil dessen, was bei ähnlich umfassender Untersuchung mit tatsächlichen Modellen aufzuwenden wäre. Außerdem bleibt eine Dynamik, die zur Selbstzerstörung führen könnte, in einer Simulation am Computer ohne Folgen (zum Beispiel Simulation einer Kernschmelze in einem Kernreaktor). In einer Simulation wird eine künstliche Welt geschaffen, die nur nach den zuvor im Programm formulierten Gesetzen arbeitet und völlig unabhängig von den Gesetzen der Physik in der tatsächlichen Welt sein kann. So kann zum Beispiel der zeitliche Ablauf eines dynamischen Verhaltens erheblich gerafft und verkürzt (Zeitraffer) oder falls notwendig, erheblich gedehnt werden (Zeitlupe). Dies macht eine genauere Beobachtung möglich. Ein anderer großer Vorteil von Computersimulationen ist sicher auch der, dass der Rechner sehr schnell Simulationen mit unterschiedlichen Ausgangswerten der Zustandsgrößen durchlaufen und so rasch die verschiedene Szenarien präsentieren kann. So wurde mithilfe einfacher Computersimulationen im praktischen Teil dieser Arbeit die optimalen Startparameter und Materialeigenschaften für die physikalische Struktur einer kinetischen Skulptur festgestellt, worauf aber dann erst im letzten Kapitel nochmal im Detail eingegangen werden soll.

Bei dieser Form der Beobachtung muss berücksichtigt werden, dass durch die Abstraktion und Modellbildung immer bestimmte Qualitäten des realen Systems verschwinden, die nicht formalisierbar sind. Wenn man zum Beispiel einen Betrachter bittet, die Qualität einer Bewegung zu beschreiben, wird er dies mit anderen Begriffen tun als denen, die bei der formalen Modellbeschreibung verwendet werden. So spricht der Betrachter zum Beispiel von der Anmut, der Eleganz oder Hektik einer Bewegung, während bei der Formalisierung für die Simulation nur noch diskrete Werte für Masse, Kraft und Beschleunigung verwendet werden, die dann in einem numerischen Lösungsverfahren die Bewegung zu beschreiben versuchen.

2.5 STRUKTUR UND VERHALTEN

Ein System kann über seine Struktur, also seine in Wechselbeziehung zueinander stehenden Elemente und deren Eigenschaften beschrieben werden. Ein andere Möglichkeit der Beschreibung stellt sein Verhalten dar, d.h. die im Zeitverlauf beobachtbare Entwicklung des Systems. Struktur und Verhalten stehen in keinem

einfachen Verhältnis zueinander. So kann ein einfach aufgebautes System durchaus ein komplexes Verhalten zeigen, sowie eine kompliziert aufgebaute Systemstruktur sich in einem einfachen Systemverhalten niederschlagen. Deshalb wird zwischen struktureller und dynamischer Komplexität unterschieden.

Ein Beispiel für ein strukturell einfaches aber dynamisch kompliziertes System ist das Doppelpendel. Damit ein System komplexes Verhalten zeigen kann, muss mindestens eine Wechselwirkung zwischen den Systemelementen nicht-linear sein und wenigstens ein Rückkopplungskreis vorhanden (vgl. Kratky 1991). Bei solch einem System kann zukünftiges Verhalten mathematisch nicht mehr abgeleitet werden, da die Zustandsgrößen exponentiell auf Störungen reagieren. Auch führen geringste Veränderungen der Anfangsbedingen nach kurzer Zeit zu einem völlig anderen Verhalten – man spricht hier auch von chaotischem Verhalten und nicht-linearen Systemen. Das Prinzip eines Doppelpendels wird im praktischen Teil der Arbeit dazu verwendet, unvorhergesehenes und überraschendes Verhalten in einer kinetischen Skulptur zu generieren. Dies soll aber später nochmals im Detail erläutert werden.

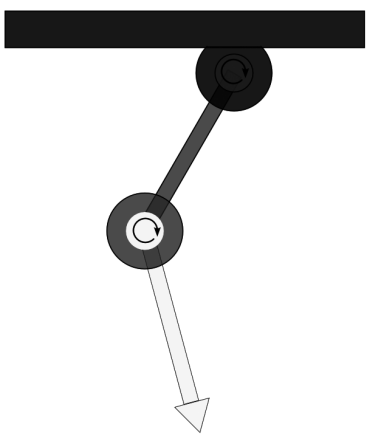


Abb. 9 Schematische Darstellung Doppelpendel

So hatte Jules Henri Poincaré bereits Ende des 19. Jahrhunderts in der Astronomie bei der Beobachtung mehrerer Himmelskörper nicht-deterministisches Verhalten erkannt. Im mechanistischen Weltbild und ihrer klassischen Physik ging man davon aus, dass alle kausalen Verhältnisse linear gelöst werden können (vgl. S. 13). Betrachtet man zum Beispiel nur die Erde und den Mond, kann die

Bahn des Erdtrabanten aus der Wechselwirkung der beiden Körper berechnet und eindeutig gelöst werden. Wenn nun aber die Sonne auch noch in die Berechnung miteinbezogen wird, kommen wir zum so genannten „Drei-Körperproblem“. So konnte Poincaré beweisen, dass selbst ab einer Anzahl von nur drei Körpern keine algorithmischen Lösungsansätze existieren. Deshalb ist es nicht möglich, Ort und Geschwindigkeit der drei Körper für jeden beliebigen Zeitpunkt zu bestimmen. Nur durch nichtlineare Gleichungen lässt sich dies darstellen, wobei diese praktisch nicht eindeutig, sondern nur näherungsweise lösbar sind. Aufgrund dieser Erkenntnis von Poincaré, sowie weiteren Arbeiten wie zum Beispiel von Edward Lorenz oder Werner Heisenberg (vgl. Trogemann und Viehoff 2005, S. 393f) musste sich die Wissenschaft Anfang des zwanzigsten Jahrhunderts endgültig vom Gedanken verabschieden, dass die Zukunft eines physikalischen Systems exakt berechnet werden kann, wenn die Anfangsbedingungen präzise feststehen.

Deshalb sind solche dynamischen und chaotischen Systeme eigentlich nicht mehr berechenbar. Hier erweist sich aber der Einsatz des Computers von großem Nutzen, da mit mit den gleichen Programmen, aber unterschiedlichen Eingabewerten sehr schnell verschiedene Szenarien beschrieben werden können (vgl. S. 31f).

„(...) wir kennen heute auf Grund dieser Theorie (Anm. d. Autors.: gemeint ist hier die Chaostheorie) die Nebenbedingungen sehr viel besser, unter denen solch ein System in chaotische Zustände geraten kann. Durch das Studium entsprechender Modellierungen komplexer Systeme, sind wir eher in der Lage, die Nebenbedingungen zu erkennen, durch die das komplexe System Wirtschaft oder das soziale System in unkontrollierbare, trudelnde Zustände kommen kann. Tatsächlich ist es ja auch so, dass wir durch das Studium etwa der unterschiedlichen Trajektorien die genauen Verlaufsformen der komplexen Systeme in begrenzten Zeitabschnitten beschreiben können.“ (Mainzer 1997)

3
PROGRAMME ALS PROZESSENTWURF

3.1 PROGRAMM ALS KONZEPT IN DER KUNST

Sind Programme bei Computern genaue Befehle, welcher Schritt nach dem nächsten von der Maschine ausgeführt werden soll, können Programme auch allgemein als Handlungsanweisungen für jegliche Art von Prozessen gesehen werden.

„Software is the programming material which any system uses, i.e. in a computer it would be the flow charts or subtractions for the computer programm. In effect software in „real“ terms is the mental intelligence required for any experience. It can also be descriped as the knowledge required for the performance of any task or transmission of communication. (...) In many cases an object is much less value than the software concerning object.“
(Levine, zitiert nach Burnham 1970, S. 60f)

Dies ist auch ein möglicher Ansatz in der Kunst, in dem das Konzept und der Prozess hinter dem Kunstwerk gleichwertig oder sogar wichtiger wird, als das tatsächliche Endergebnis – diese Form der Kunst verschiebt ihren Fokus vom Objekt zum Gedankensystem dahinter und wird auch als Konzeptkunst bezeichnet. In dem Text „Paragraphs of Conceptual Art“ (1967) nimmt der Künstler Sol Lewitt Bezug auf seine Kunst, in deren Fokus vor allem das Konzept steht.

„In conceptual art the idea or concept is the most important aspect of the work. When an artist uses a conceptual form of art, it means that all of the planning and decisions are made beforehand and the execution is a perfunctory affair. The idea becomes a machine that makes the art.“ (LeWitt 1967)

Hier finden wir auch wieder Parallelen zu den Handlungsanweisungen eines Computerprogramms. So haben wir schon im ersten Teil dieser Arbeit gesehen (vgl. S. 19), dass durch eine Trennung der Maschine von ihrem Programm – eine

wichtige Voraussetzung dafür, dass eine Maschine überhaupt erst programmiert werden kann – die Software immer wichtiger ist, als die dahinter liegende Hardware. Ob diese Handlungen nun von einem Computer oder auch von einem Menschen ausgeführt wird, ist auch zweitrangig, was ebenfalls schon mit Turings Beispiel der so genannten „Papiermaschine“ gezeigt wurde (vgl. S. 22f). Trotzdem ist das endgültige und physische Objekt sehr wichtig, weil ohne ein solches würde das Konzept nur in der imaginären Welt der Gedanken bleiben. Erst dadurch, dass der Wille zur tatsächlichen Form wird, kann der Betrachter die Gedanken des Künstlers auf gewisse Art und Weise wahrnehmen. An dieser Stelle finden wir Parallelen zu den Programmen bei Rechenmaschinen. So schlägt auch hier der Programmtext erst um in Handlungen und Prozesse, wenn dieser von einer Maschine ausgeführt wird.

Abb. 10 Sol Lewitt, „minimal art apple“

LeWitt macht deshalb darauf aufmerksam, dass die Form eines Objekts in seiner Kunst zwar eine untergeordnete Rolle spielt, trotzdem aber sehr wichtig für die Grammatik des Werkes ist.

„The form itself is of very limited importance; it becomes the grammar for the total work. In fact, it is best that the basic unit be deliberately uninteresting so that it may more easily become an intrinsic part of the entire work. Using complex basic forms only disrupts the unity of the whole. Using a simple form repeatedly narrows the field of the work and concentrates the intensity to the arrangement of the form.“ (LeWitt 1967)

Sein Fokus beim tatsächlichen Werk liegt deshalb mehr in der Anordnung der einzelnen Elemente eines Objektes und deren Beziehungen zueinander im Raum, als deren eigentlichen Aussehen. Ein Ansatz der auch im praktischen Teil dieser Arbeit seine Anwendung findet. Dieses Konzept der Verwendung von meist sehr puren Materialien kann sicher auch als Weiterentwicklung der Tendenzen der abstrakten Malerei Anfang des letzten Jahrhunderts gesehen werden. So gingen die Vertreter der abstrakten Kunst davon aus, dass das, was den Betrachter eines Gemäldes tatsächlich anspricht, nicht die Wiedergabe eines erkennbaren Gegenstandes ist, sondern die Form und Farbgebung, die Texturen und die Linienführung an sich. Und auch Vertreter der so genannten „Softwarekunst“ nahmen die abstrakte Kunst oft als Vorbild für ihre Werke.

3.2 FORMGESTALTUNG VS. PROZESSGESTALTUNG

Wie schon gezeigt, bleibt beim Computer die eigentliche Struktur einer Maschine meist verborgen und die Ausführung von Programmanweisungen wird nicht mehr wahrgenommen (vgl. S. 23).

„(...) ‚Software Art‘ ist eine Reportage über den Code. Denn Softwarekunst ist keinesfalls Kunst für Programmierer. Softwarekunst ist die Lesbarmachung des Codes.“ (Gohlke 2003, S.4)

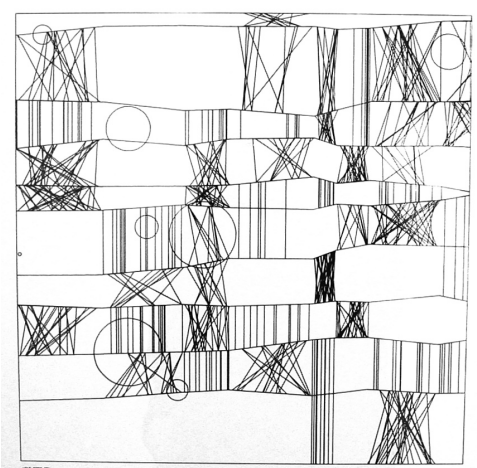


Abb. 11 Frieder Nake, 13/9/65 Nr. 2 „Hommage à Paul Klee“

Dadurch, dass in der Softwarekunst die einzelnen Programmanweisungen wieder in wahrnehmbare, meist visuelle Elemente übersetzt werden, werden dieser zuvor unsichtbaren inneren Vorgänge der Maschine, wieder für den Betrachter sichtbar – allerdings muss hier berücksichtigt werden, dass solch eine Transformation natürlich nie direkt sein kann, sondern immer eine freie Interpretation des Künstlers ist. Pioniere der Computergrafik wie A. Michael Noll, Georg Ness, Herbert W. Franke und Frieder Nake benutzten Programme mit ihrer zugrunde liegender Mathematik als eine Art Notenschrift für künstlerische Bilder. In ihrer Softwarekunst der 60er Jahre experimentierten sie mit abstrakten Grundformen um hier erste Erfahrungen mit dem neuen Medium zu sammeln. Grundprinzipien von gemalten Bildern wurden herausgearbeitet (sozusagen ihr „Skelett"), um dann in Regeln für den Ablauf von Programmen formuliert zu werden – mit dem Ergebnis, neue Bilder mithilfe des Computers zu generieren (vgl. Franke 2007). Die endgültige Form erhält das Kunstwerk erst über die Ausführung des Programms durch die Maschine. Programmierung wird zum Akt des Entwerfens.

Auch in der Architektur gibt es Ansätze, in denen der Prozess, also das Programm zur Formfindung im Fokus der Gestaltung steht und nicht das finale Objekt. Dies macht vor allem dann Sinn, wenn zwar das Problem und die Kriterien für eine erfolgreiche Lösung bekannt sind, nicht aber die finale Antwort. Eine Strategie kann dabei die Anwendung von Genetischen Algorithmen sein. Hier wird versucht, evolutionäre Prinzipien, wie Selektion, Rekombination und Mutation so weit zu modellieren, dass sie auf Computern simuliert werden können. Durch diese evolutionären Mechanismen soll über den Simulationsprozess eine Optimierung des Modells stattfinden. Wie bereits erwähnt funktioniert dies aber nur, wenn sowohl die Aufgabe formal beschreibbar ist, als auch die Rahmenbedingungen und das Ziel exakt definiert werden können. So verwendeten John Frazer und Peter Graham genetische Algorithmen für die Entwicklung eines Schiffsrumpfs einer Rennyacht.

„A population was generated with very small genetic mutations. Each hull (Anm. d. Autors.: engl. Ausdruck für „Schiffsrumpf") was then plottet out and complex calculations were performed for displacement, trim, waterplane, wetted surface and for block and prismatic coefficients which gave some indication of the likely performance of the hull. Natural selection was applied by breeding further populations from the hulls with the most promising coefficients."
(Frazer 1995, S. 61)

Dieses Beispiel zeigt sehr schön, wie Programme dazu genutzt werden können, Teil des Gestaltungsprozesses zu werden. So wird weniger die eigentliche Form

modelliert als wie vielmehr der Prozess, der die Form generiert.

„We are describing processes, not components; ours is the backet-of-seeds as opposed to the bag-of-bricks approach"
(Frazer 1995, S. 99)

Der Algorithmus wird zum physischen Material, jedoch erfolgt die natürliche Selektion hier immer noch durch den Menschen.

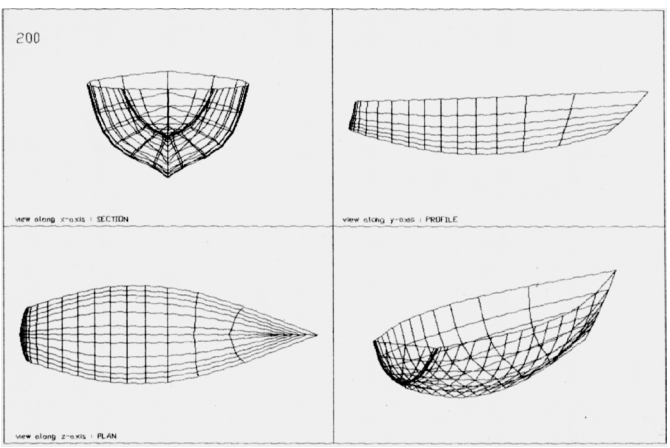


Abb. 12 Frazer und Graham, Entwicklung eines Schiffrumpfs

In der grafischen Gestaltung wird meist ein Raster als Hilfskonstruktion zur Anordnung von grafischen Elementen verwendet. Hier besteht ein Raster in der Regel aus der Aufteilung der Bildfläche in ein System von X Reihen mal Y Spalten. Dieses Gerüst hilft während des Gestaltungsprozesses bei der Ordnung der grafischen Elemente – es definiert sozusagen das Regelwerk, innerhalb dessen die visuellen Elemente ihre Größe und Position durch den Gestalter erhalten können. Wird allerdings mit dynamischen System gearbeitet ist diese klassische Rastermethode alleine nicht mehr sinnvoll. Beobachten wir aktuelle Tendenzen der Entwicklungen von zeitbasierten Medien, wie zum Beispiel der computerbasierten Erzeugung von Bewegtbildern oder der Gestaltung von interaktiven Anwendungen, so sehen wir, dass dieses starre Gerüst kaum mehr zeitgemäß ist. Ein Ansatz ist es, dass klassische zweidimensionale Raster um die Dimensionen der Zeit und des Raums zu erweitern. Aber wie verhalten sie die einzelnen Elemente eines solchen dynamischen Systems über die Zeit? So besteht eine Möglichkeit darin,

schon vorab durch das Setzen von Keyframes auf einer Zeitleiste exakt festzulegen, was der dynamische Charakter eines Elementes ist. Alternativ zu dieser Methode können aber auch Verhaltensregeln für die grafischen Objekte definiert werden. Diese beschreiben, in was für einem Rahmen sie sich zu verhalten haben und auf welche Impulse sie wie reagieren sollen.

„A very elegant way of developing grid systems for moving pictures sequences is by determining the rules and the behaviours to be assigned to objects and graphic elements. Following this principle, the designer who uses behaviours becomes a producer who determines the form and behaviour of his or her scene and the ‚actors'."
(Diezmann und Gremmler 2003, S. 67)

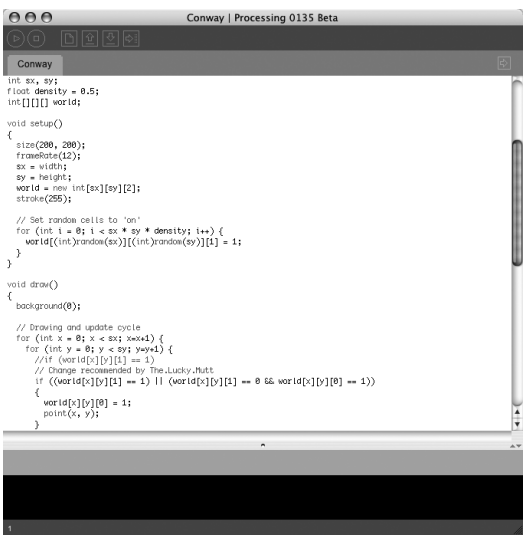


Abb. 13 Entwicklungsumgebung „Processing"

Dies ermöglicht das Gestalten von komplexen Sequenzen, ohne mühsam und zeitintensiv einzelne Keyframes auf einer Zeitleiste setzen zu müssen. Außerdem kann es bei dieser Form eines Gestaltungsprozesses auch immer wieder zu überraschenden und inspirierenden Momenten kommen, da die Programmanweisungen erst bei der Ausführung durch den Computer sichtbar werden. Projekte wie die freie Entwicklungsumgebung „Processing" (www.processing.org) von Ben Fry und Casey Reas, versuchen Künstler und Gestalter an das Schreiben von Compu-

terprogrammen heranzuführen. Hier werden spannende Möglichkeiten eröffnet, auch als Nicht-Entwickler mit diesen prozessorientierten Gestaltungskonzepten zu experimentieren. Auch für die Entwicklung der Simulationen für den praktischen Teil dieser Arbeit wurde „Processing“ verwendet, da hier sehr schnell nützliche Ergebnisse erzielt werden können.

3.3 TATSÄCHLICHE BEWEGUNG ALS AUSFÜHRUNG VON PROGRAMMEN

In der kinetischen Kunst wird das künstlerische Konzept zur tatsächlichen Raumskulptur mit ihren beweglichen Elementen. Schon für Beuys galt die Plastik als Überführung von etwas Metaphysischen in die Stoffeswelt (vgl. Beuys 1970).

Abb. 14 Tim Prentice, „Yellow Zingers“

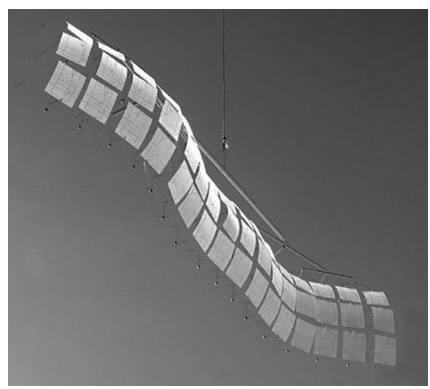


Abb. 15 Tim Prentice, „Pequoit Maquette“

Und bei der kinetischen Kunst sind es neben der Skulptur vor allem die Bewegungen der einzelnen Elemente, die zu den „formgewordenen“ Gedanken des Künstlers werden. So beschreibt schon Aristoteles die Bewegung als ontologisches Prinzip, das das Seiende vom Nichtseienden trennt. Bewegung ist für ihn jener Prozess, der die Bedingung des Inerscheinungstretens der Dinge beinhaltet (vgl. Buderer 1992, S.9). Und erst über die physikalische Materialiät der Skulpturelemente wird die Bewegung sichtbar für den Betrachter.

Werden Programme auch allgemein als Handlungsanweisungen für jegliche Art von Prozessen gesehen werden (vgl. S. 37) so finden wir diesen Ansatz auch in der Mechanik für die Bewegungsabläufe dieser Maschinen wieder – die Struktur der Maschine, d.h. wie die einzelnen Elemente zueinander verbunden sind, wird

zum Programm für die Bewegung. So waren in der „Softwarekunst“ diese Regeln noch im Programmtext versteckt und wurden erst über eine subjektive Transformation seitens des Künstlers für den Betrachter sichtbar. In einigen kinetischen

Abb. 16 Ralf Baecker, „Rechnender Raum“

Skulpturen, wie zum Beispiel den windbetriebenen Objekten eines Tim Prentice, sind nun diese Regeln durch physikalische Verbindungen der einzelnen Elemente zueinander definiert und damit auch für den Betrachter sichtbar. Allerdings können solche tatsächlichen Verknüpfungen eine so hohe Komplexität erreichen, dass das Zusammenspiel der einzelnen Elemente für den Betrachter kaum mehr nachvollziehbar ist. Dies kann aber dann wieder den eigenen Reiz einer Arbeit ausmachen, wie Ralf Baecker sehr schön bei seiner Arbeit „Rechnender Raum“ zeigt.

So sah der Künstler Jean Tinguely seine mechanischen Skulpturen unter anderem auch als Gegenentwurf zu der Technologie unserer Zeit.

„Die moderne Technik ist anonym, allumfassend und diskret geworden. Letzteres auch, indem sie das Rad, und die zirkuläre Bewegung, etwa beim Computer, völlig zu verbergen vermag, während meine Plastiken gerade auf diesem Prinzip, auf Rad und Kreisbewegung beruhen. Aber weil Technik geräuschlos geworden ist und sich mit dem Design, mit der glatten Schale und der Stromlinienform maskiert, macht sie uns vergessen, dass wir von ihr beherrscht werden.“ (Tinguely 2000, S. 65f)

Abb. 17 Jean Tinguely, „Matrac“

Und wer schon einmal ein Werk Tinguelys tatsächlich in Bewegung gesehen hat, weiß, dass es unter anderem dieses stetig sichtbare, „nicht-perfekte“ und meiste auch mit großem Lärm verbundene Ausführen der einzelnen Schritte ist, was den besonderen Charme seiner Maschinen ausmacht.

„Ich baue in sich freie Maschinen, die ihre eigene anarchistische Freiheit, ihr eigenes Chaos, ihre Unordnung und Ordnung haben und auf ihre eigene Weise ihren Zufall erzeugen.“ (Tinguely 2000, S. 65f)

Die soll auch wieder im praktischen Teil dieser Arbeit aufgegriffen werden.
Die zeitliche Abfolge von beweglichen Teilen in kinetischen Skulpturen kann auch als Rhythmus bezeichnet werden. Wie in der Musik gibt er auch hier die zeitliche Abfolge der einzelnen Elemente vor. So wird bei den kinetischen Skulpturen des Künstlers Nicolas Schöffers der Rhythmus des optischen und akustischen Verhaltens als ständige Reaktion auf die Umwelt festgelegt. Damit spiegelt sich sich in dem ästhetischen Rhythmus der Plastik der des Raumes, in den das Objekt in-

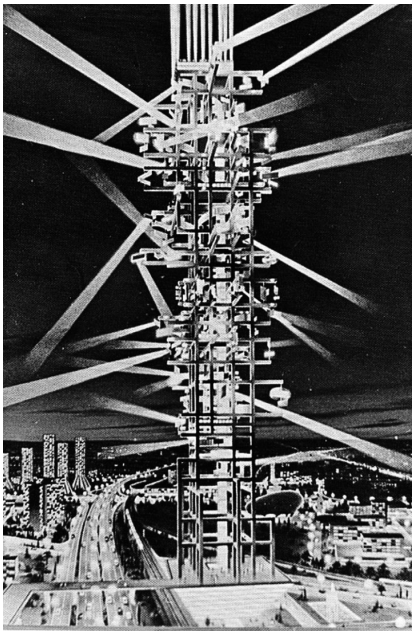


Abb. 18 Nicolas Schöffer, „Tour Lumière Cybernétique“

tegriert ist. Er sieht seine Werke auch als kybernetische Systeme und arbeitet deshalb auch mit den Prinzipien des Regelkreises (vgl. S. 28). Für das nie realisierte Projekt eines kybernetischen Turmes in Paris 1968 waren für einen näheren Umkreis Mikrophone, photoelektrische Zellen, Thermometer, Hygrometer und Anemometer vorgesehen, um Informationen aus dem Umfeld zu messen. Der 307 Meter hohe Turm sollte bewegliche Spiegel und Scheinwerfer besitzen, die sich dann in Abhängigkeit zur ihrer gemessenen Umwelt bewegen konnten.

Er ist damit auch seiner Zeit weit voraus und wir finden heute wieder ähnliche Konzepte bei den medialen Fassaden von Gebäuden und reaktiven Skulpturen (vgl. Bullivant 2005). Für Schöffer sind seine Werke auch immer mit einem dazugehörigen künstlerischen Programm verbunden, welches ihm auch wichtiger ist, als das eigentliche Objekt (vgl. Buderer 1992, S. 196). Der erste Schritt – von ihm auch als „Phase de la Création“ (Schöffer 1961, S. 90f) bezeichnet – bestimmt welche Parameter aus der Umwelt als Informationsgeber in sein System aufgenommen werden. Schöffer bezeichnet sie als Unbestimmtheiten (Indeterminationen), da nicht immer vorhersehbar ist, wie stark sie auftreten werden. Der nächste Schritt ist das Festlegen von optischen und akustischen Anamorphosen, d.h. es werden

Verhaltenssequenzen vorgegeben, wie sich zum Beispiel ein Element bewegen soll. Erst in einer zweiten Phase wird die Struktur – von Schöffer als „Construcion Rigoureuse“ bezeichnet (ebenda, S. 91) – festgelegt, in der die Anamorphosen den Indeterminationen zugeordnet werden. Da seine Systeme trotz der unvorhersehbaren Indeterminationen immer wieder zur Ordnung tendieren, baut er bewusst künstliche Störungen ein

„Denn in jedem kybernetischen System, das das Werk programmiert, gibt es eine Störzelle. Diese Zelle wird zu einem bestimmten Zeitpunkt zufällig das ganze Programm stören. Dann beginnt das Werk von neuem mit der Erstellung eines neuen Programms. Die Zelle ist ein Störfaktor, der die Beziehungen zwischen den Impulsempfängern und den Informationsgebern stört. Der Störfaktor ist im System eingebaut: die Made ist praktisch im Apfel.“
(Schöffer, zitiert nach Buderer 1992, S. 191)

Somit reagiert eine solches System nicht nur auf seine Umwelt sondern immer wieder auch auf sich selbst – ein Ansatz der, wie wir später sehen werden, für die Umsetzung der kinetischen Skulptur der vorliegenden Arbeit sehr wichtig ist.

4
FAZIT

Computerprogramme stehen mit ihren algorithmischen Lösungsverfahren in direkter Tradition zu der rationalen Vorgehensweise in den klassischen Naturwissenschaften. So wurde in diesem mechanistischen Weltbild angenommen, dass wenn alle Anfangsbedingungen eines zu betrachteten Ausschnitts der realem Umwelt präzise feststehen, die Zukunft von diesem exakt bestimmt werden kann. Die Forschung bezog sich daher ausschließlich auf die messbaren Aspekte ihrer Beobachtungen und konnte nun auch aus ihrer früheren Machtlosigkeit gegenüber den Naturkräften heraustreten. Diese Abstraktion bedeutete aber auch immer eine Reduktion des Ganzen, da nunmehr alle Erfahrungstatsachen unberücksichtigt blieben, die nicht in den jeweiligen Begriffsrahmen passten. Allerdings war es erst auf Basis dieser Vereinfachung möglich, Modelle und Experimente zu formulieren, um zuvor aufgestellte Theorien beweisen zu können. Allerdings ist ein solches Modell letzten Endes immer ein anderes Objekt, als der Gegenstand, den es modelliert und es gehorcht exakt den Verhaltensgesetzen, die von einer entsprechenden Theorie explizit formuliert sind oder aus ihr abgeleitet werden können. Verstehen wir Programme bei Computern als festgelegtes Regelwerk für den internen Ablauf, so bedeutet ein Programm zu schreiben auch, einer Welt Gesetze zu geben. Diese Regeln werden vom Programmierer festgelegt und haben ihre eigenen Bedingungen in ihrer eigenen Welt. Dieses unterschiedliche Verhalten von Abläufen in der Realität und ihrer rechnergestützten Simulation spielte auch im Gestaltungsprozess der praktischen Arbeit eine große Rolle.

Funktionierten die klassischen Maschinen noch nach dem Vorbild des arbeitenden Armes, so konnten mit der Entwicklung der ersten Rechenmaschinen nun auch Prozesse ausgeführt werden, die zuvor dem menschlichen Geist vorbehalten waren. Neben den Fortschritten in der handwerklichen Mechanik war vor allem die rationalistische Tradition eines mechanistischen Weltbildes und die damit einhergehende Abstraktion und Formalisierung wissenschaftlicher Ergebnisse eine wesentliche Voraussetzung für diese Entwicklung. Auch wenn das „Drei-Körperproblem“ von Poincaré zeigt, dass es Probleme gibt, die nicht eindeutig gelöst werden können, stehen diese ersten Rechenmaschinen mit ihrer weiteren Ent-

wicklung zum heutigen Computer in direkter Tradition zu einem deterministischen Lösungsansatz. Erst durch Abstraktion und Formalisierung können algorithmische Verfahren zur Lösung eines Problems beschrieben werden. Ein solches Verfahren setzt bei einer Verarbeitung durch eine Maschine voraus, dass diese nur eine endliche Anzahl von Stellungen, auch „Zustände“ genannt, einnehmen kann. So gab es bei den ersten mechanischen Rechenmaschinen noch unkontrollierte Schwankungen, da ein solcher Zustand nie einer endlichen Anzahl von Stellungen entsprechen konnte. Mit der Einführung der elektronischen Rechner konnte nun die Genauigkeit bei Zuständen kontrolliert werden. Der Prozessor kann nun hier die Maschine in diskrete Zustände bringen. Dieses digitale Funktionsprinzip hat aber nun wieder den Nachteil, dass Zufall eigentlich vom einem Computer nicht generiert werden kann – auch war er für diese Maschinen nie vorgesehen, da sie ja in direkter Tradition zu den deterministischen Verfahren in den klassischen Naturwissenschaften stehen. Hier bieten die unkontrollierten Schwankungen von analogen, mechanischen Systemen wieder eine interessante Alternative. Ein Ansatz der auch später in der Umsetzung der kinetischen Skulptur der praktischen Arbeit seine Anwendung finden soll.

Wenn wir einen Algorithmus als Lösungsverfahren begreifen, in welchem schon zu Beginn alle Eingabewerte feststehen, so zeichnen sich interaktive Systeme dadurch aus, dass sie zur Laufzeit neue Eingaben verarbeiten können – die Folge können nicht-lineare Programmabläufe sein. Feedbackmechanismen wie im Regelkreis der Kybernetik bieten hier interessante Ansätze zur Steuerung und Regelung solcher nicht-linearen dynamischen Systeme. So zeichnet sich ein Regelkreis dadurch aus, dass das Ergebnis eines Vorgangs die Ausgangsbedingungen beeinflusst und wieder steuert. Somit hängt die Zukunft eines Prozesses nicht nur von externen Systemeingaben ab, sondern in gleichem Maße auch von der eigenen Vergangenheit des Systems. Ein Ansatz, der auch für dramaturgische Abfolge der später beschriebenen kinetischen Skulptur von großer Bedeutung ist. Hier werden diese Mechanismen zum einem durch einfache Programme auf Mikrocontrollern, aber auch durch die eigentliche Materialität der Skulptur gesteuert. Ein Punkt der später nochmal im Detail aufgezeigt werden soll.

Programme sind in erster Linie Handlungsanweisungen für die Ausführung von bestimmten Schritten. So schlägt beim Computer die Schriftform des Codes erst durch die Ausführung der Maschine um in Handlungen und Prozesse. Wenn wir die archetypischen Arbeitsmachinen im Gottkönigtum des Pyramidenzeitalters betrachten, so konnten die Visionen der Herrscher auch erst durch die mühsame Arbeit ihrer Untertanen umgesetzt werden. Wichtig war hier, dass Menschenpotenzial zu konzentrieren und vor allem die Organisation zu disziplinieren, um eine effektive Ausführung zu erreichen. Auch Turing zeigt bei seiner „Papiermaschine“,

dass ein Mensch mit Papier, Bleistift und Radiergummi, sowie strikter Disziplin ausreicht, die Arbeit einer Rechenmaschine im gleichen Maße auszuführen. Neben den Handlungsanweisungen eines Programms braucht es also auch immer einen Aktivitätsträger, der diese ausführt – andernfalls bleibt das Programm ein rein theoretisches Konstrukt. Verstehen wir in der kinetischen Kunst das Konzept des Künstlers als eine Form von Programm, so werden die beweglichen Elemente erst über ihre Materialität sichtbar. Schon für Beuys galt die Plastik als Überführung von etwas Metaphysischen in die Stoffeswelt. So wird auch bei der kinetischen Skulptur der praktischen Arbeit das künstlerische Konzept zum physischen Material. Die Handlungsanweisungen des Programms werden zum Bewegungsablauf der einzelnen Elemente des realen Objektes.

Programmierbare Maschinen zeichnen sich dadurch aus, dass ihr konkretes Verhalten geändert werden kann, ohne nochmals etwas an ihrer Konstruktion zu verändern. Dies setzt eine Trennung von Hardware und Software voraus. So zeigte Turings Konzept einer universell-programmierbaren Maschine, dass jede Maschine, deren Hardware ein bestimmtes Minimum an Kompliziertheit und Flexibilität erreicht, jede andere Maschine durch spezielle Programmanweisungen simulieren kann. Dies bedeutet, dass die eigentliche Maschine immer weniger wichtig wird als das Programm, welches auf ihr abläuft. So sind nur noch die Ein- und Ausgaberelationen und die internen Zustandsübergänge einer Maschine entscheidend. Die Maschine als solche verliert immer mehr ihren eigentümlichen Charakter zu Gunsten ihrer universellen Funktionalität. Zudem wurde die beim digitalen Rechner eingesetzte Technologie über die Zeit immer kompakter und schneller und wir nehmen in der Ausführung der Programme die physikalischen Übergänge von einem Zustand in einen anderen direkt gar nicht mehr wahr – die Maschine wird immer mehr zur anonymen Black Box für den äußeren Betrachter. Softwarekunst kann zwar als Sichtbarmachen des internen Codes verstanden werden, allerdings ist hier die visuelle Ausgabe immer eine subjektive Übersetzung der unsichtbaren inneren Vorgänge der Maschine. Verstehen wir Programme auch als allgemeine Handlungsanweisungen für jegliche Art von Prozessen, finden wir hier wieder interessante Ansätze in der kinetischen Kunst. Sind in der Softwarekunst die Regeln für den Ablauf noch im Code versteckt, so können diese für den Betrachter bei einer kinetischen Skulptur wieder sichtbar gemacht werden. So wird auch die Struktur der kinetischen Skulptur in der praktischen Arbeit wieder Teil des Programms für den Ablauf der Bewegungen.

5
VORSTUDIEN

5.1 AUSGANGSPUNKT

Allgemein betrachtet sind Programme zunächst einmal eine Folge von Befehlen für die Ausführung von bestimmten Schritten. Schreiben wir Programme für den Computer, so bedeutet dies vor allem seine eigene Gedanken zu abstrahieren und formalisieren. Durch diesen Schritt der Vereinfachung und formalen Beschreibung können die Zeichen nun von einer Maschine weiterverarbeitet werden. Erst in der tatsächlichen Ausführung eines Programmtextes durch den Computer – der Mechanisierung – schlägt der Text dann um in Verhalten. Programmierung ist deshalb auch immer Begegnung mit sich selbst und mit den eigenen formulierten Gedanken und Vorstellungen. Werden selbst geschriebene Programme zur visuellen Gestaltung verwendet, können hier komplexe Ergebnisse trotz einfacher Verhaltensregeln erzielt werden (vgl. S. 41f). Solche Programme haben zu Beginn auch meist einen sehr experimentellen Charakter und der Gestalter erhält durch das Verändern einzelner Parameter im Programm eine Vielzahl von visuellen Ergebnissen. Dadurch, dass das Ergebnis auch erst mit der Ausführung durch die Maschine sichtbar wird, passieren hier auch immer wieder nicht geplante Ereignisse, die dann aber wiederum Inspiration für neues Schaffen sein können. Das Ergebnis solcher generativen Gestaltungsprozesse wird dann meist auf dem Bildschirm des Computers ausgegeben, wobei diese Ausgabe aber immer nur eine visuelle Übersetzung des internen Programmablaufs darstellt (vgl. S. 39f). Die eigentlichen Programmanweisungen, die für die Ausgabe verantwortlich sind bleiben dem Betrachter verborgen. Kinetische Objekte wie von Tim Prentice (vgl. S. 43) zeigen nun diese Verhaltensregeln wieder. Die physikalische Struktur des Objektes definiert den Handlungsspielraum der Arbeit – das Programm wird zum tatsächlichen und sichtbaren Bewegungsablauf der Skulptur. Dieser Ansatz soll auch im praktischen Teil dieser Arbeit weiterverfolgt werden.

Wie in den vorangegangen Kapitel gezeigt, stehen Computerprogramme mit ihren algorithmischen Lösungsverfahren in direkter Tradition zu der rationalen Vorgehensweise im mechanistischen Weltbild. Zufall war hier nie vorgesehen und kann

von einer elektronischen Maschine eigentlich auch nie wirklich erzeugt werden, da sie nur eine endliche Anzahl von Zuständen einnehmen kann. Digital erzeugter Zufall – der deshalb ja eigentlicher keiner ist – wird immer wieder in einer Vielzahl von Computeranwendungen, wie zum Beispiel bei Simulationsprozessen, Spielen oder genetischen Algorithmen eingesetzt. Dadurch wird eine gewisse Nähe zur Realität geschaffen. Bei realen kinetischen Objekten muss nun aber dieser Zufall nicht simuliert werden. In den mechanisch erzeugten Bewegungsabläufen der Objekte existieren schon von Natur aus unkontrollierte Schwankungen. Diese können auch explizit zum Thema gemacht werden wie Tinguelys Skulptur „Matrac“ (vgl. S. 45) zeigt und ein wichtiger Teil für den dramaturgischen Ablauf der Arbeit werden. Diese Möglichkeiten der analogen Zufallserzeugung sollen auch in der praktischen Umsetzung dieser Arbeit experimentell erforscht werden.

5.2 ERSTE EXPERIMENTE

Bei den ersten Versuchsaufbauten zu dieser Arbeit wurde der Computer zuerst bewusst aus dem Gestaltungsprozess ausgeschlossen. Vielmehr galt es anhand von einfachen analogen Modellen erste Erfahrungen für die spätere kinetische Skulptur zu sammeln. Auch war zu Beginn nicht klar, was das endgültige Ergebnis der Arbeit sein würde. Vielmehr sollten erst im Experiment mögliche Umsetzungen gefunden werden.

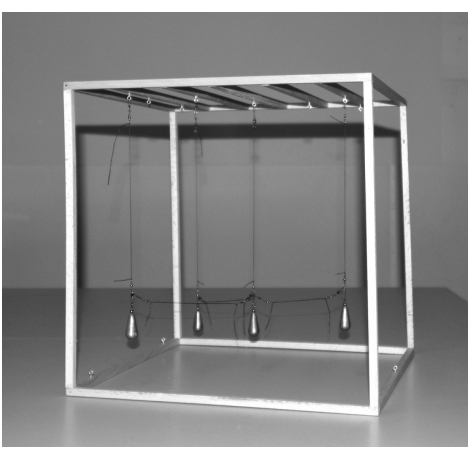


Abb. 19 Modell mit Gewichten

Frühe Experimente beschäftigten sich mit den Möglichkeiten, wie einzelne Elemente über den Einsatz von physikalischen Materialien zueinander in Beziehung gestellt werden konnten. Bei einem ersten Versuchsaufbau wurden in einem kleinen Hängekasten Gewichte angebracht und durch Gummischnüre miteinander verbunden. Dieses System von gekoppelten Schwingungen erbrachte zunächst interessante Ergebnisse im Bewegungsablauf. Wird ein Element manuell in Schwingung versetzt, überträgt es seine Eigenfrequenz auf die verbundenen Glieder. Jeder Nachbar versucht nun wiederum seine individuelle Grundschwingung anzunehmen. Dadurch dass jede Bewegung eines Elements Auswirkungen auf das gesamte System hat, entstehen in diesem einfachen Feedbacksystem erste chaotische Abläufe.

Neben dem manuellen Erzeugen von Kräften wurde auch in einem weiteren Schritt mit dem Einsatz von künstlichen Impulsgeneratoren experimentiert. Hierfür wurden die Gewichte im zuvor beschriebenen System durch kleine Zugmagnete ersetzt, welche über einen Mikrocontroller angesteuert werden konnten. Erhält ein Zugmagnet eine elektrische Spannung wird im Inneren ein Eisenbolzen von einem Elektromagneten angezogen und eine ruckartige Verschiebung des eigenen Schwerpunktes ist die Folge. Diese Funktion kann wiederum zur Erzeugung von Impulsen im zuvor beschriebenen System genutzt werden. Über ein näherungsweises Verfahren wurde im nächsten Schritt versucht, die Frequenz dieses Impulses auf die Eigenfrequenz des Elements anzupassen um eine positive Rückkopplung auf die Schwingung zu erzielen.

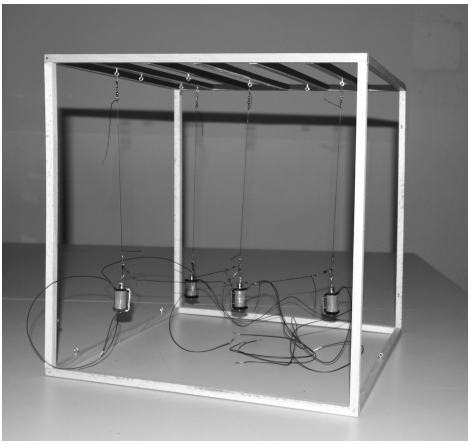


Abb. 20 Modell mit Zugmagneten

Alternativ zur manuellen Ermittlung der Eigenfrequenz wurden auch Piezoelemente benutzt, um eine exakte Messung der eigenen Schwingung durchzuführen. Diese Sensoren wurden zwischen Aufhängung und Gummischnur angebracht und nach Verstärkung des Signals konnte am Analogeingang eines Mikrocontrollers exakt die Grundschwingung eines Elements ermittelt werden. Durch einen einfachen Feedbackmechanismus im Programmablauf des Mikrocontrollers konnte nun die Schwingung des Zugmagneten optimal angeregt werden.

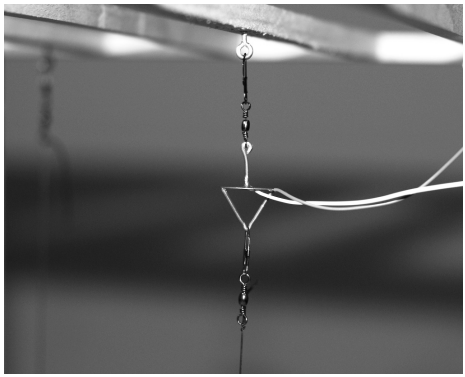


Abb. 21 Piezoelement zur Messung

Im Anschluss an diese ersten Experimente mit tatsächlichen Materialien entstand auch eine Computersimulation mithilfe derer einfache dynamische Systeme erstellt und simuliert werden konnten. Der Benutzer kann hier im zweidimensionalen Raum Elemente platzieren und miteinander verbinden. Die einzelnen Parameter für eine spätere physikalische Simulation können festgelegt und auch während der Simulation zur Laufzeit verändert werden.

Abb. 22 Parameter Computersimulation

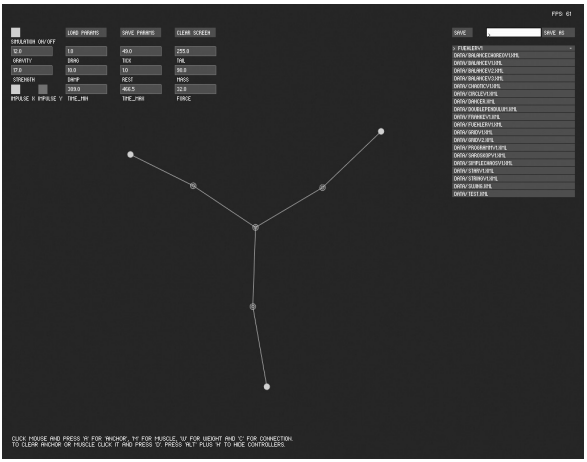


Abb. 23 Bespiel dynamisches System in Computersimulation

Wird ein zufriedenstellendes Resultat erreicht, kann dieses dann abgespeichert und später wieder aufgerufen werden. Der große Vorteil einer solchen Simulation gegenüber analogen Modellen in der Realität ist, dass sehr schnell verschiedene Szenarien ohne den Bau von aufwendigeren analogen Modellen durchgespielt werden können. In einer Simulation wird eine künstliche Welt geschaffen, die nur nach den zuvor im Programm formulierten Gesetzen arbeitet. In diesem System haben Elemente ihre eigenen Bedingungen in ihrer eigenen Welt. So kann zum Beispiel der zeitliche Ablauf des Prozesses erheblich gedehnt werden, was eine genauere Beobachtung der Auswirkung verschiedener Parametereinstellungen ermöglicht (vgl. S. 32). Dieser Vorteil kann aber auch wieder zum Nachteil werden. So konnten zwar mithilfe der Simulation interessante Ergebnisse erzielt werden, jedoch beim Versuch diese in die reale Welt zu übertragen, verhielten sich diese dynamischen Systeme meist anders als in ihrer künstlichen Umgebung und waren deshalb auch weniger zufriedenstellend. Dies hatte zur Folge, dass bei den nächsten Arbeitsschritten der Computer vorerst nicht eingesetzt werden sollte. Vielmehr lag nun der Fokus wieder beim Bau tatsächlicher Modelle und dem Experiment mit verschiedenen Materialien.

Nach dem Bau eines größeren Hängekasten konnten nun sehr schnell einzelne Elemente befestigt und miteinander verbunden werden. Verschiedene Materialien wie Gewichte, Gummibänder, Schwingfedern, Federstahldraht und Seile wurden hierzu eingesetzt und auf ihre Eigenschaften hin überprüft. Schnell zeigte sich, dass hier das Verhalten der einzelne Elemente zueinander stark vom ver-

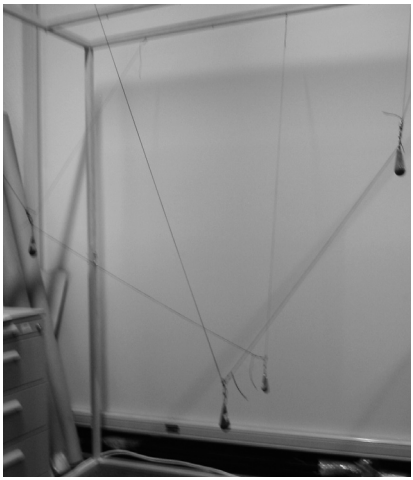


Abb. 24 Testen Materialeigenschaften

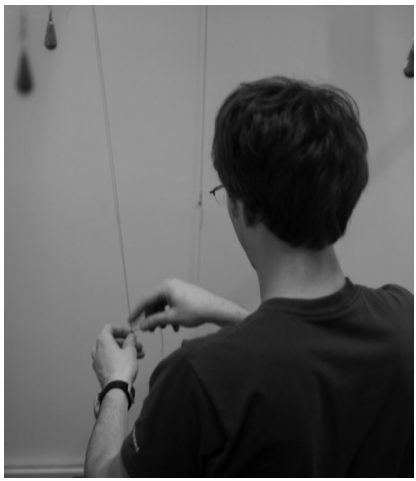


Abb. 25 Knüpfen von Strukturen

wendeten Material und den räumlichen Dimensionen abhing. Wie in früheren Modellen wurden auch hier neben dem manuellen Erzeugen von Kräften automatisierte Mechanismen eingesetzt. Mithilfe von Schrittmotoren, die sich als besser geeignet zeigten als Zugmagnete konnten nun Gewichte über einen einfachen Mechanismus in Bewegung versetzt werden.

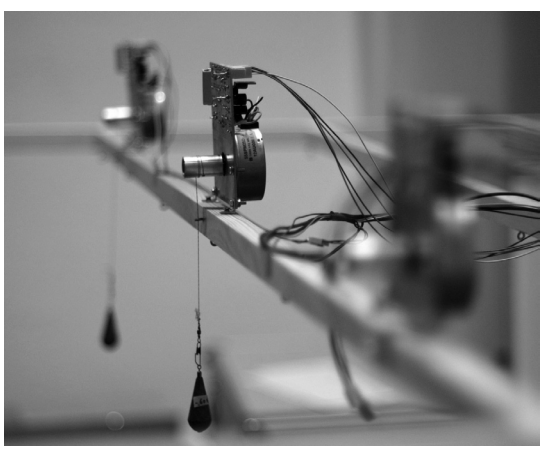
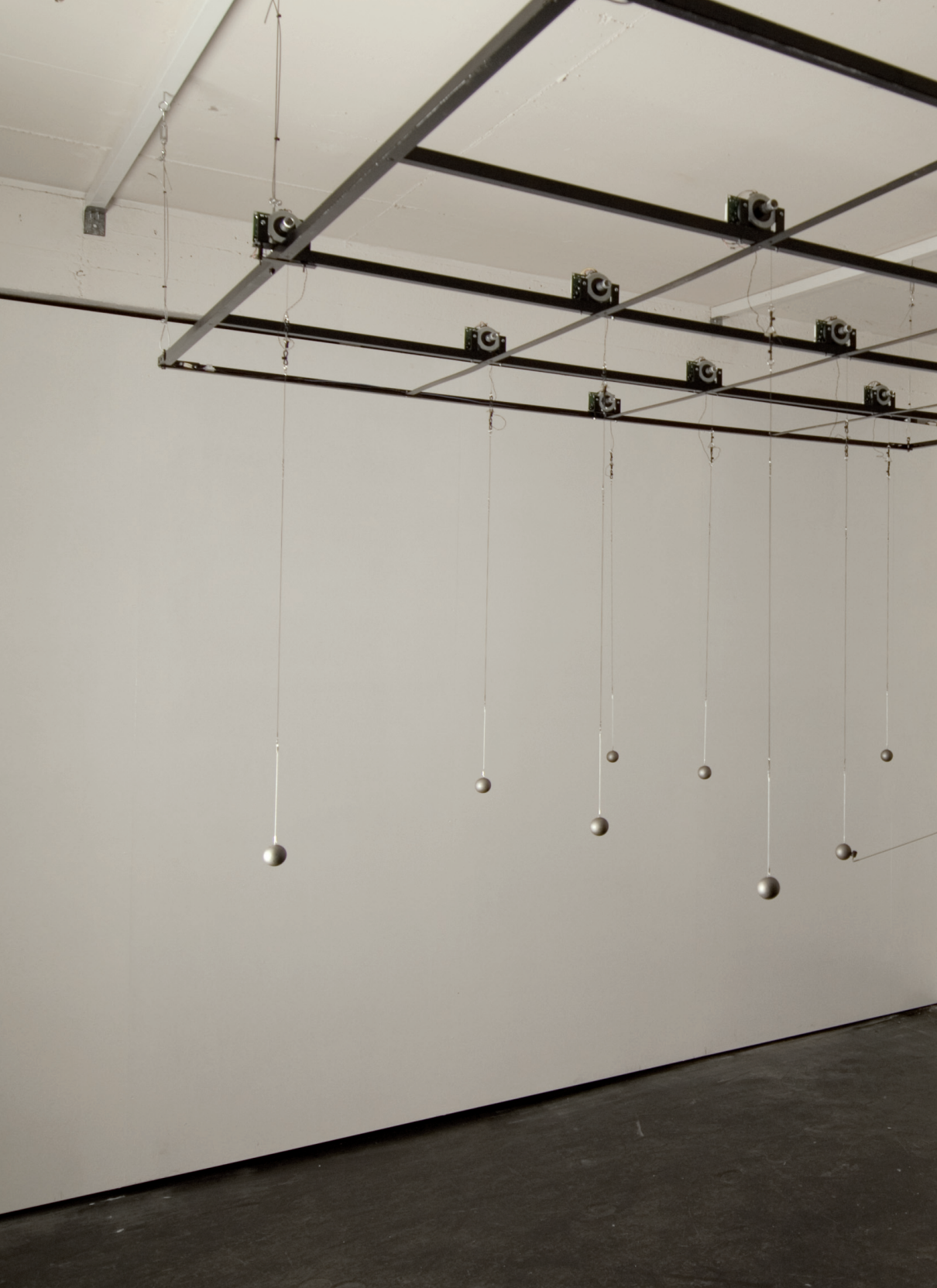
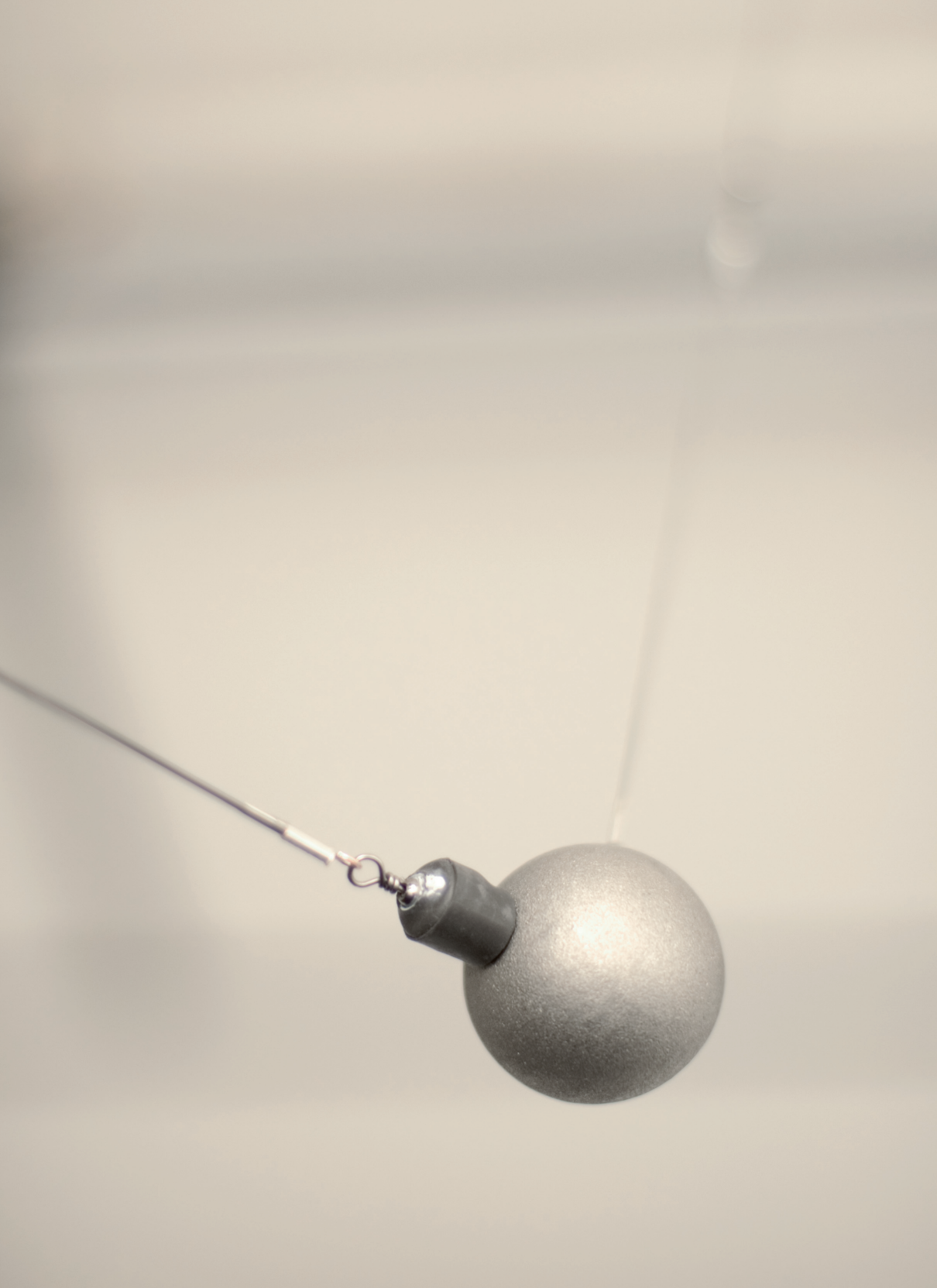
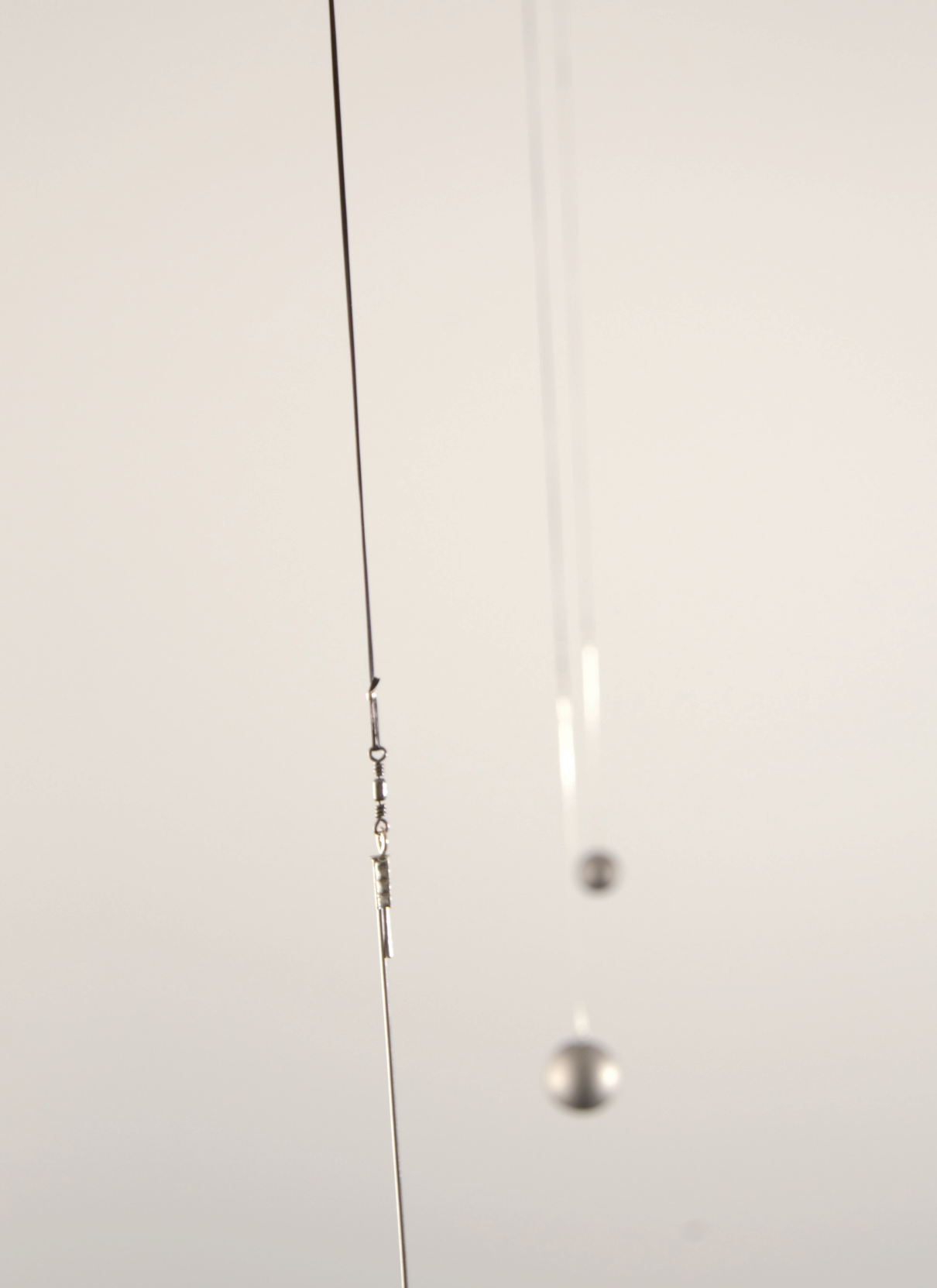


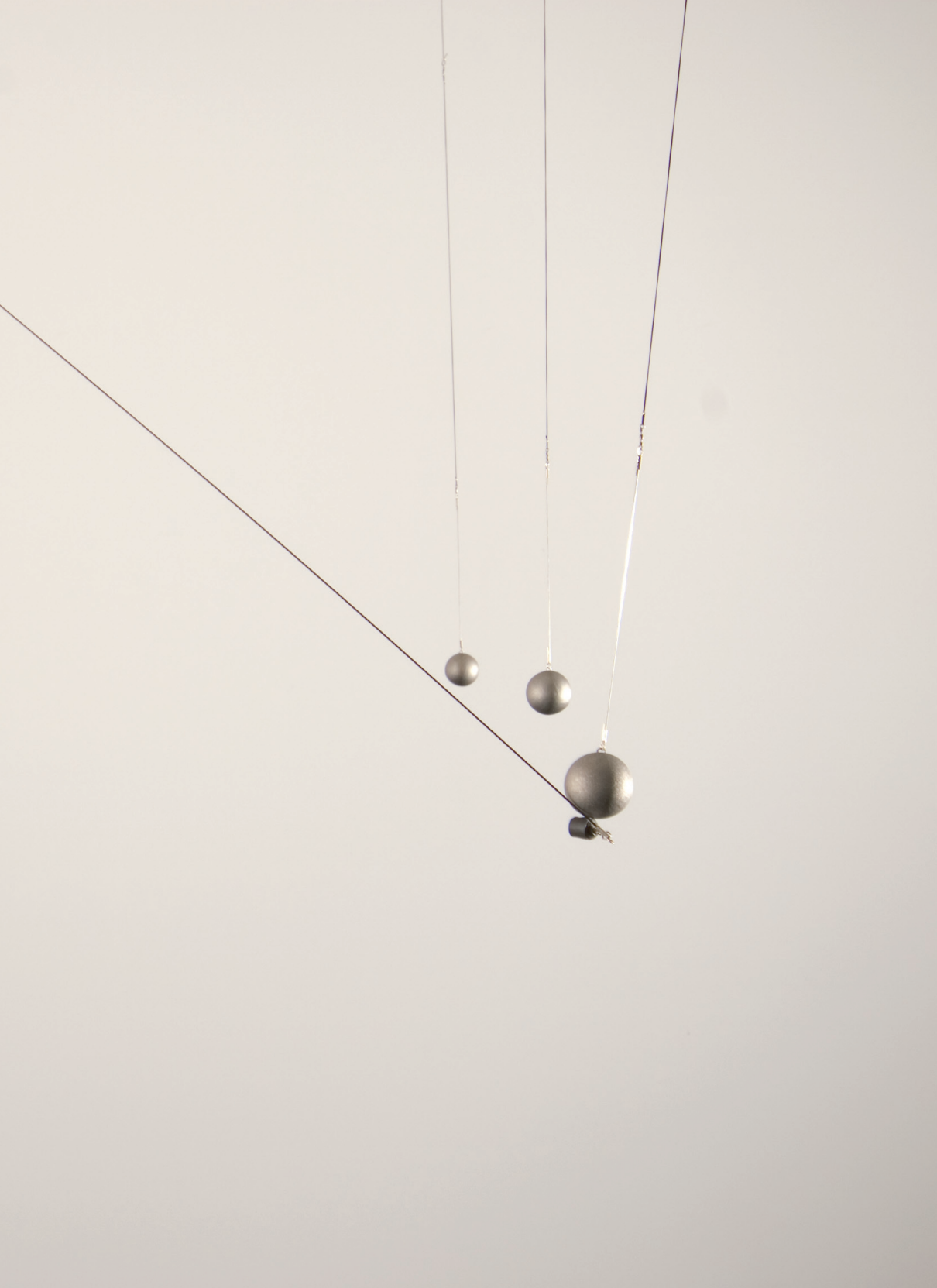
Abb. 26 Einsatz von Schrittmotoren

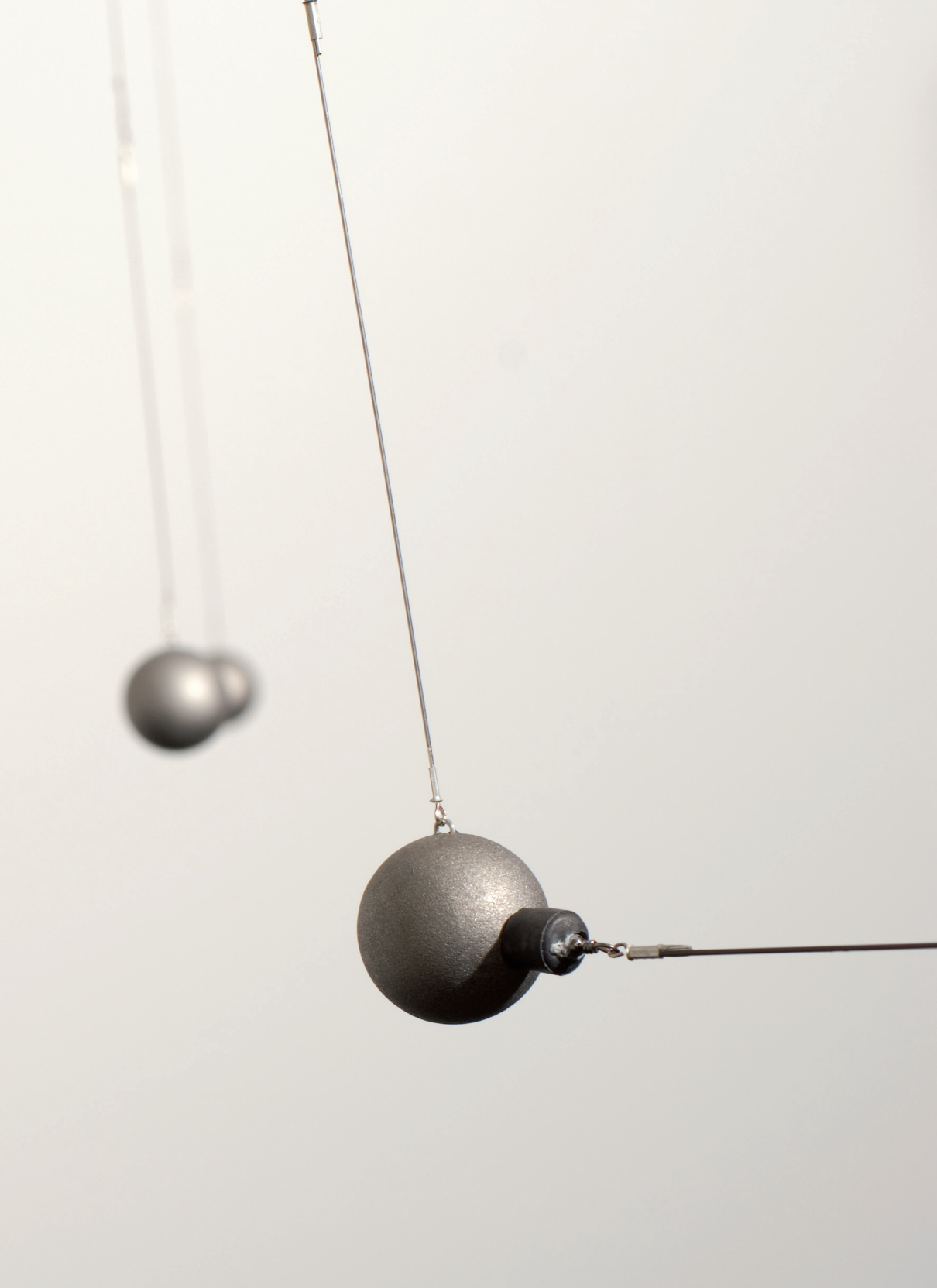
Hierbei oszillierten die einzelnen Elemente jeweils um ihren Schwerpunkt in der Aufhängung, die eigentliche Struktur des Systems blieb aber unverändert. Bei früheren Arbeiten mit rechnergestützten Gestaltungsprozessen waren es immer auch Fehler beim Ablauf, die zur Inspiration für neues Schaffen führten – und auch hier sollte eine Fehlfunktion einen Denkanstoß für den weiteren Verlauf der Arbeit geben. Im realen Versuchsaufbau riss die Verbindung zwischen zwei Elementen an einem Punkt und verhakte sich an einem anderen Glied. Das System veränderte auf diese Weise während dem Prozess seine eigene Struktur – ein Verhalten dass zu einem wichtigen Mechanismus bei der finalen Umsetzung werden sollte.











6
CONNECT

Eigenreaktive kinetische Skulptur

6.1 BESCHREIBUNG

Die Skulptur „Connect“ besteht aus dreizehn beweglichen Eisenkugeln, die über Gummischnüre in einer Matrix angebracht sind. Ein physikalischer Impulsträger steuert das Verhalten der einzelnen Elemente zueinander. Jedes Element besitzt sein eigenes Programm zur Steuer- und Regelung – es existiert kein Hauptprogramm, dass das System von außen steuert. Ist ein Modul mit dem Impulsträger verbunden regt es seine eigene Schwingung solange über einen Motor an, bis sich dieser Träger löst und mit einem anderen Element verbindet. In einem kontinuierlichen Prozess bildet die Skulptur somit ihre eigenen Strukturen und löst diese wieder auf. Dadurch, dass ein solches analoges System den ständigen Störungen seiner realen Umwelt ausgesetzt ist, entstehen hierbei komplexe Bewegungsmuster, die in ihrer Abfolge nicht absehbar sind.

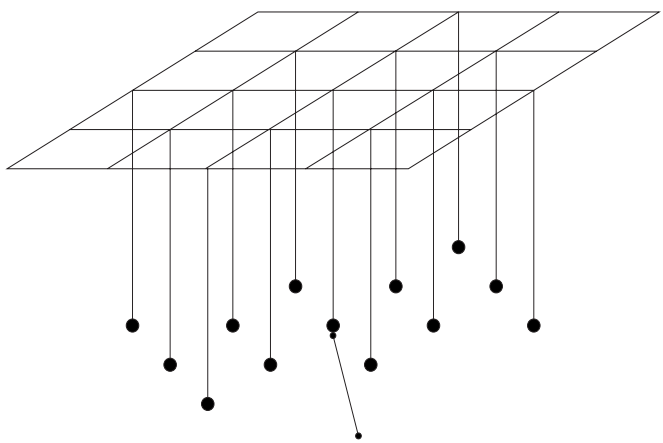


Abb. 27 Schematischer Aufbau „Connect“

6.2 AUFBAU

Jedes der dreizehn Module besteht aus einem Schrittmotor an dem über eine Gummischnur eine Eisenkugel befestigt ist. Zwischen Schnur und Motor ist ein Piezoelement angebracht und über einen Mikrocontroller am Motor kann somit das eigene Schwingverhalten festgestellt und optimal angeregt werden. Diese positive Rückkopplung hätte in der Theorie einen Systemzusammenbruch zur Folge (vgl. S. 30). Im Modul findet aber über die Materialeigenschaft der Gummischnur eine ständige Dämpfung statt.

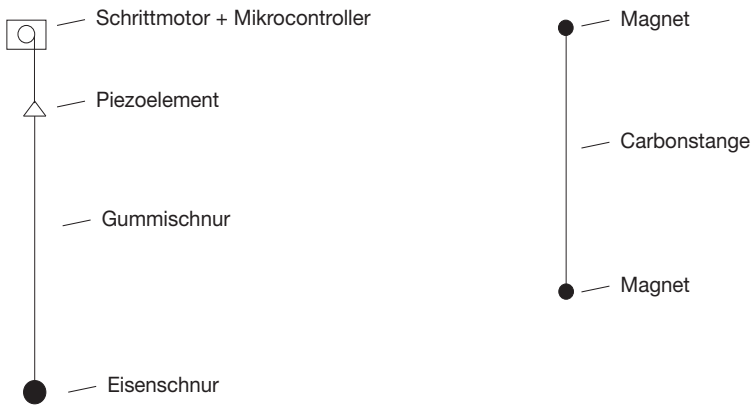


Abb. 28 Aufbau eines Moduls und Aufbau des physikalischen Impulsträgers

Der physikalische Impulsträger, der für das Bewegungsverhalten der Skulptur verantwortlich ist besteht aus einer Carbonstange mit jeweils einem Ringmagneten an jedem Ende. Über diese Magneten kann sich das Element an einem Modul einhaken.

Ist ein Modul mit dem Impulsträger verbunden beginnt es seine eigene Schwingung zu messen und über den Schrittmotor optimal anzuregen. Die Struktur, die durch die Verknüpfung entsteht entspricht der eines Doppelpendel (vgl. S. 33). Hier reagiert das Schwingverhalten sehr sensibel auf geringste Störungen durch die Umwelt und eine nicht-lineare Bewegung ist die Folge. Deshalb ist es nie möglich, vorauszusehen, wie lange dieser Mechanismus braucht bis sich der Träger mit einem anderen Modul verbindet. Wichtig war hierbei, dass keine Elektromagnete verwendet werden, sondern das Verbinden und Lösen ausschließlich über die Bewegung der Skulptur ausgeführt wird.

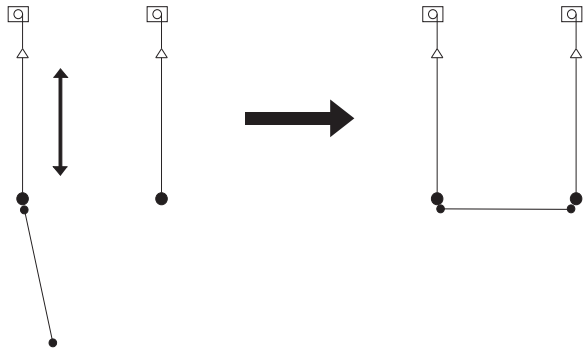


Abb. 29 Mechanismus zum Knüpfen einer neuen Verbindung

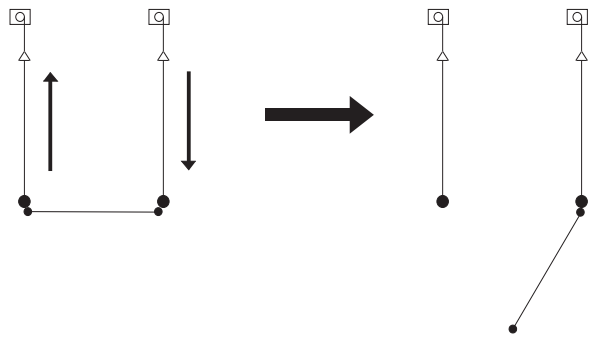


Abb. 30 Mechanismus zum Lösen der alten Verbindung

Existiert eine Verbindung zwischen zwei Modulen versuchen beide ständig ihre Eigenfrequenz über eine Messung optimal anzuregen. Schwingen die Elemente disharmonisch zueinander entstehen hier aufgrund eines gekoppelten Federsystems enorme Kräfte und die magnetische Verbindung reißt an einem der beiden Aufhängepunkte. Das Modul, das nicht mehr verbunden ist stoppt die Anregung seiner Eigenfrequenz. Somit ist der physikalische Impulsträger auch immer verantwortlich dafür, wo Bewegung in der Skulptur stattfindet. Auch kann sein Weg durch die Skulptur nicht vorhergesehen oder gar berechnet werden.

Die Steuereinheit eines Moduls basiert auf der freien Physical Computing Plattform „Arduino“ (www.arduino.cc). Neben einem Mikrocontroller befinden sich auf jeder Platine Elemente für die Regelung des Schrittmotors und die Messung des

Piezoelements. Dadurch, dass die einzelnen Platinen speziell für diese Arbeit konzipiert wurden, mussten sie im Eigenverfahren bestückt und gelötet werden.

Für die interne Steuerlogik ist es sehr wichtig herauszufinden, ob ein Modul mit dem Impulsträger verbunden ist oder nicht. Hierzu wird die Schwingung der Eisenkugel über das zuvor beschriebene Verfahren angeregt und anschließend eine Messung am Analogeingang des Mikrocontrollers durchgeführt. Es stellte sich heraus, dass wenn die Amplitude über mehrere Schwingungen aufgezeichnet wurde der Durchschnitt dieser Werte einen brauchbaren Indikator zur Zustandsermittlung darstellt. Ist ein Modul mit einem Impulsträger verbunden ergibt sich hier ein kleinerer Wert als wenn dies nicht der Fall ist. Dies lässt sich dadurch erklären, dass der Impulstträger durch das zuvor beschriebene Prinzip eines Doppelpendels ständig Störungen im eigenen Schwingverhalten erzeugt.

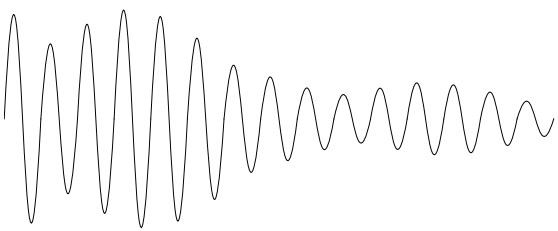


Abb. 31 Messung der Amplitude ohne Verbindung

Abb. 32 Messung der Amplitude mit Verbindung

Die eigentliche Logik des Programms auf dem Mikrocontroller ist deshalb recht einfach. Da ein solches Programm sich eigentlich immer am besten durch sich selber erklärt, soll hier auf eine detaillierte Beschreibung verzichtet werden und nur ein grober schematischer Aufbau gezeigt werden.

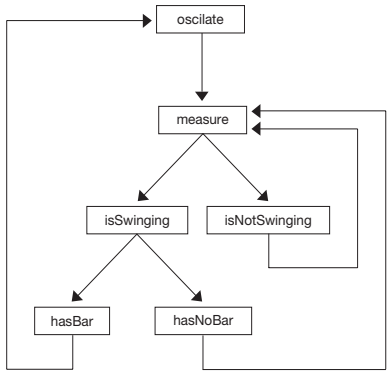


Abb. 33 Logik Programmablauf auf Mikrocontroller

Dadurch, dass das Programm auf Messungen der Umwelt reagiert – zum Beispiel ob ein Modul mit einem physikalischen Impulsträger verbunden ist oder nicht – wird auch die physikalische Struktur der Skulptur zu einem Teil von diesem. Auf diesen Punkt wird in der abschließenden Schlussfolgerung noch einmal genauer eingegangen.

Für die Ermittlung der optimalen Anordnung der Module im Raum wurde neben einem analogen Versuchsaufbau auch eine rechnergestützte Simulation eingesetzt. Hier bot vor allem die Simulation den großen Vorteil, dass sehr schnell verschiedene Szenarien durchgespielt werden konnten.

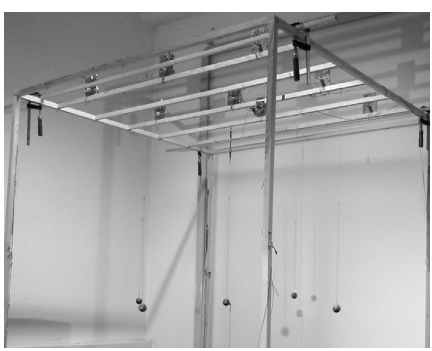


Abb. 34 Analoger Versuchsaufbau

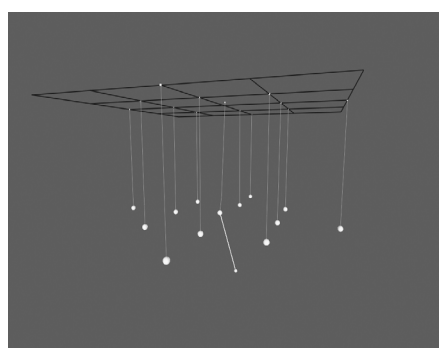


Abb. 35 Simulation

Schnell zeigte sich, dass die Anzahl der Module eine gewisse Menge nicht überschreiten sollte, da ansonsten der Fokus des Betrachters auf die Grundfunktion des Verknüpfens und Lösens von neuen Strukturen verloren geht. Aus diesem Grund fand eine Beschränkung auf ein 5x5 Raster statt. Dieses Raster, welches gleichzeitig auch die Aufhänpunkte für Module definiert stellt gleichzeitig ein visuelles Regelwerk dar, wie wir es aus der grafischen Gestaltung kennen (vgl. S. 41). Hier einige mögliche Anordnungen:

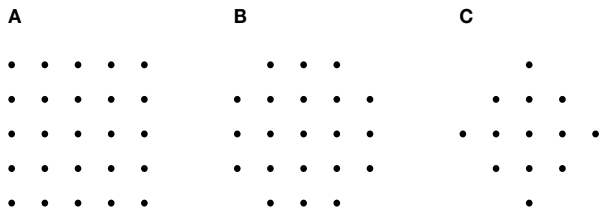


Abb. 36 Mögliche Anordnungen

Variante „C“ zeigte in der Simulation, dass hier die Chance einer raschen Verknüpfung in der Mitte der Struktur sehr hoch war und nach außen immer geringer wurde. Diese Eigenschaft ist sehr interessant für den dramaturgischen Ablauf der Arbeit. Deshalb kommt diese Form der Anordnung auch in der finalen Umsetzung zum Einsatz.

6.3 CONCLUSIO

Die Arbeit „Connect“ erzeugt komplexes Verhalten trotz einfachster Regeln in den Unterprogrammen jedes Moduls. Trotz Einschreibens der Verhaltensregeln in den Speicher der einzelnen Module findet über das physische Zusammenspiel aller Elemente der Skulptur ein Kontrollverlust beim Prozessieren des gesamten Ablaufs statt. Es findet keine klassische Trennung mehr von Hardware und Software statt. Die Programme auf den Mikrocontrollern wissen nichts über ihre eigentliche Umwelt. Die Messung der Schwingung ist ihre einzige Möglichkeit nach der realen Welt zu tasten. Erst über den analogen Impulsträger in Form eines mobilen Carbonstabes können tatsächliche Verbindungen hergestellt und wieder gelöst werden. Dieser Verbindungsmechanismus wird nicht elektronisch gesteuert, sondern erfolgt ausschließlich über das Bewegungsverhalten der einzelnen Module. Somit wird die sich ständig ändernde Struktur der Skulptur Teil des eigentlichen Programmablaufs. Die Grundregel – wann automatisierte Bewegung stattfindet

und wann nicht – wird nicht verborgen, sondern ist mit dem Impulsträger ständig für den Beobachter sichtbar. Zwar sind in den Steuereinheiten der einzelnen Module nur eine bestimmte Anzahl von Zuständen definiert, allerdings entstehen über die tatsächliche physikalische Struktur nicht kontrollierbare Schwankungen, was einen chaotischen Ablauf zur Folge hat. Dadurch kann auch nicht vorhergesehen werden zu welchem Zeitpunkt eine Verbindung hergestellt, beziehungsweise wieder gelöst wird. Somit wird ein ständiger Spannungsbogen im Ablauf der Arbeit erzeugt und es entstehen überraschende Momente beim Beobachter des Werkes.

7
ABBILDUNGEN

Abb. 1 Archetypische Arbeitsmaschine, Aus: Trogemann und Viehoff 2005, S. 50.

Abb. 2 Skizze der Rechemaschine Schickards, Aus: Trogemann und Viehoff 2005, S. 59.

Abb. 3 Zählen mit Händen und Füßen, Aus: Ifrah 1991, S. 39.

Abb. 4 Schema einer speziellen Turingmaschine:
Aus: http://www.uni-bielefeld.de/idm/neuro/Denken_in_Symbolen.html

Abb. 5 EDVAC:
Aus: http://commons.wikimedia.org/wiki/Image:Edvac.jpg

Abb. 6 Schema der Analytical Engine,
Aus: http://www1.cs.columbia.edu/~zeph/3261/home.html

Abb. 7 Regelkreis in der Kybernetik

Abb. 8 Positives und Negatives Feedback
Aus: Trogemann und Viehoff 2005, S. 417.

Abb. 9 Schematische Darstellung Doppelpendel
Aus: http://de.wikipedia.org/wiki/Doppelpendel

Abb. 10 Sol Lewitt, „minimal art apple“
Aus: http://radio.weblogs.com/0109045/categories/berlinerSkulpturen/

Abb. 11 Frieder Nake, 13/9/65 Nr. 2 „Hommage à Paul Klee„
Aus: http://container.zkm.de/presse/secure/Algo_300dpi/Nake_Klee.jpg

Abb. 12 Frazer und Graham, Entwicklung eines Schiffrumpfs
Aus: Frazer 1995, S. 61.

Abb. 13 Entwicklungsumgebung „Processing“

Abb. 14 Tim Prentice, „Yellow Zingers", Aus: http://www.timprentice.com/

Abb. 15 Tim Prentice, „Pequoit Maquette“, Aus: http://www.timprentice.com/

Abb. 16 Ralf Baecker, „Rechnender Raum“, Aus: http://www.no-surprises.de

Abb. 17 Jean Tinguely, „Matrac“,
Aus: http://container.zkm.de/presse/special_algorithmische.html

Abb. 18 Nicolas Schöffer, „Tour Lumière Cybernétique“,
Aus: http://www.olats.org/schoffer/biograph.htm

Abb. 19 Modell mit Gewichten

Abb. 20 Modell mit Zugmagneten

Abb. 21 Piezoelement zur Messung

Abbildungen

Abb. 22 Parameter Computersimulation
Abb. 23 Bespiel dynamisches System in Computersimulation
Abb. 24 Testen Materialeigenschaften
Abb. 25 Knüpfen von Strukturen
Abb. 26 Einsatz von Schrittmotoren
Abb. 27 Schematischer Aufbau „Connect“
Abb. 28 Aufbau eines Moduls und Aufbau des physikalischen Impulsträgers
Abb. 29 Mechanismus zum Knüpfen einer neuen Verbindung
Abb. 30 Mechanismus zum Lösen der alten Verbindung
Abb. 31 Messung der Amplitude ohne Verbindung
Abb. 32 Messung der Amplitude mit Verbindung
Abb. 33 Logik Programmablauf auf Mikrocontroller
Abb. 34 Analoger Versuchsaufbau
Abb. 35 Simulation
Abb. 36 Mögliche Anordnungen

8
LITERATUR

BATESON, Gregory, Ökologie des Geistes, Frankfurt am Main 1985.

BEUYS, Joseph, Provokation – Lebensstoff der Gesellschaft. Zu Kunst und Antikunst, 1. Aufzeichnung des 67. Forumgesprächs der Reihe „Meinung gegen Meinung“, Podium: Max Bense, Joseph Beuys, Max Bill, Arnold Gehlen, Leitung: Wieland Schmied. Veranstalter: ASG Arbeitsgemeinschaft Sozialpädagogik und Gesellschaftspädagogik, Düsseldorf 1970.

BOSSEL, Hartmut, Modellbildung und Simulation – Konzepte, Verfahren und Modelle zum Verhalten dynamischer Systeme. Vieweg Verlag, Braunschweig 1994.

BUDERER, Hans-Jürgen, Kinetische Kunst – Konzeption von Bewegung und Raum, Wernersche Verlagsgesellschaft, Worms 1992.

BULLIVANT, Lucy, 4dspace: Interactive Architecture, In: Architectural Design, Vol 75 No 1 Jan/Feb 2005, London 2005.

BURNHAM, Jack, Software: information technology: its new meaning for art, Catalogue published for the exhibition Software, Jewish Museum, New York, N.Y., U.S., September 16 to November 8, 1970 [itinerary: Smithsonian Institution, Washington, D.C., U.S., December 16, 1970, to February 14, 1971], Curated by Jack Burnham, Jewish Museum , New York 1970.

COY, Wolfgang, Analog/Digital, In: Hyperkult II – Zur Orstbestimmung analoger und digitaler Medien, Analog/Digital, Warnke, Coy, Tholen (Hrsg.), transcript Verlag, Bielefeld 2005.

DIEZMANN UND GREMMLER, Grids for the Dynamic Image, AVA Publisihing SA, Crans-près-Céligny 2003.

FRANKE, W. Herbert, Science Fiction Autor und Pionier der Computergrafik, Lesung, Präsentation und Diskussion, Aula Kunsthochschule für Medien, Köln 2007.

FRAZER, John, An Evolutionary Architecture, Architectural Association Publications, Themes VII, copyright John Frazer and the Architectural Association, London 1995.

GOHLKE, Gerrit, Software Art – Reportage über den Code, Künstlerhaus Bethanien (Hrsg.), Berlin 2003.

HAUGELAND, John, Analog und Analog, In: Analog/Digital - Opposition oder Kontinuum? Zur Theorie und Geschichte einer Unterscheidung, Böhnke und Schröter (Hrsg.), transcript Verlag, Bielefeld 2004.

HÖFLING, Oskar, Lehrbuch der Physik (Mittelstufe B), Bonn 1965.

HUXLEY, Aldous, Science, Liberty and Peace, Fellowship Publications, New York 1946.

IFRAH, Georg, Universalgeschichte der Zahlen, Campus Verlat, Frankfurt am Main, 1991.

KAPLAN, Abraham, The Conduct of Inquiry, San Franzisco 1964.
KRATKY, Karl W., Die „Beherrschbarkeit“ komplexer Systeme, In: Systemische Perspektiven. Interdisziplinäre Beiträge zu Theorie und Praxis, Kratky (Hrsg.), Carl Auer, Heidelberg 1991.
LEWITT, Sol, „Paragraphs on Conceptual Art,“ In: Artforum 5, no. 10, pp. 79–83, 1967.
MAINZER, Klaus, Interview: Komplexe Systeme, intelligente Computer und Selbsorganisation, 02/02/1997, http://www.heise.de/bin/tp/issue/r4/dl-artikel2.cgi?mode=html&artikelnr=2161.
MATTHIES, Michael, Einführung in die Systemwissenschaft, WS 2002 / 2003, 2002, http://www.usf.uos.de/cgi-bin/archive?~vberding/syswi/.
MOLES, Abraham A., Die Kybernetik, eine Revolution in der Stille, In: Epoche Atom und Automation, Enzyklopädie des technischen Zeitalters, Genf 1959.
MUMFORD, Lewis, Mythos der Maschine – Kultur, Technik und Macht, Fischer Taschenbuch Verlag, Frankfurt am Main 1977.
NAKE UND GRABOWSKI, Zwei Weisen, das Computerbild zu betrachten, In: Hyperkult II – Zur Orstbestimmung analoger und digitaler Medien, Analog/Digital, Warnke, Coy, Tholen (Hrsg.), transcript Verlag, Bielefeld 2005.
PENROSE, Roger, Computerdenken - Des Kaisers neue Kleider oder Die Debatte um Künstliche Intelligenz, Bewußtsein und die Gesetze der Physik. Orig.: The Emperor‘s New Mind. Spektrum der Wissenschaft. Heidelberg 1991.
PIAS, Claus, Zeit der Kybernetik – Eine Einstimmung, In: Erinnerungen an die Macy-Konferenzen und die Gründung des Biological Computer Laboratory (2002), In: Cybernetics | Kybernetik, The Macy-Conferences 1946-1953, Band II, Essays und Dokumente, Pias und Vogel (Hrsg.), Diaphanes Verlag, Zürich-Berlin 2004.
SCHÖFFER, Nicolas, Stucture et Indétermination – Formes ouvertes – Anamorphose optique et temporelle, In: Art International (Zschr.), S. 90/91, June/August 1961.
SCHRÖTER, Jens, Analog/Digital - Opposition oder Kontinuum?, In: Analog/Digital - Opposition oder Kontinuum? Zur Theorie und Geschichte einer Unterscheidung, Böhnke und Schröter (Hrsg.), transcript Verlag, Bielefeld 2004.
STACHOWIAK, Herbert, Allgemeine Modelltheorie, Springer-Verlag, Wien-New York 1913.
TARNAS, Richard, Idee und Leidenschaft: Die Wege des westlichen Denkens, Zweitausendeins Verlag, Frankfurt am Main 1997.
TINGUELY, Jean, L‘Ésprit de Tingueley, Hatje Cantz Verlag, Ostfildern 2000.
TROGEMANN UND VIEHOFF, Code@Art – Eine elementare Einführung in die Programmierung als künstlerische Praktik, Springer-Verlag, Wien 2005.
VON FOERSTER, Heinz, Erinnerungen an die Macy-Konferenzen und die Gründung des Biological Computer Laboratory (2002), In: Cybernetics | Kybernetik, The Macy-Conferences 1946-1953, Band II, Essays und Dokumente, Pias und Vogel (Hrsg.), Diaphanes Verlag, Zürich-Berlin 2004.
WEGNER, Peter, The Paradigm Shift from Algorithms to Interactions, Brown University, Final Draft, October/14/1996, http://www.cs.brown.edu/people/pw/.
WEIZENBAUM, Joseph, Die Macht der Computer und die Ohnmacht der Vernunft, Suhrkamp Verlag, Frankfurt am Main 1977.

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht.

Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Köln, Januar 2008

Andreas Muxel

Ich danke Martina Höfflin, Prof. Georg Trogemann und Frans Vogelaar für die ausgezeichnete Betreuung dieser Arbeit.

Großer Dank gilt auch:

Martin Nawrath, durch dessen technische Betreuung die praktische Umsetzung dieser Arbeit erst möglich geworden ist

allen Studenten und Mitarbeitern des Lab3, im besonderen Karin Lingnau, Lasse Scherffig, Ludwig Zeller, Ralf Baecker und Stefan Göllner für die interessanten und wertvollen Gespräche

Martin Hesselmeier für die tolle Unterstützung gerade am Ende dieser Arbeit

Michael Pichler, der mich bei der Gestaltung des Layouts tatkräftig unterstützt hat

Frau Dr. Christiane König für erste Gespräche, bevor diese Arbeit entstanden ist

Heinz Nink für den Aufbau der Installation

Peter Schuster und Bernd Voss für die Lichtinszenierung

Mischa Kuball für die Bereitstellung eines Raumes für die Ausstellung

Ein ganz besonderer Dank gilt meinen Eltern, ohne die dieses zusätzliche Studium nicht möglich gewesen wäre und ella für die Geduld und moralische Unterstützung während der ganzen Arbeit.