

Helge Hauptfleisch

Untersuchung eines mechanischen Oszillators zur Demonstration der Schwingungsgleichung

Staatsexamensarbeit

HD-KIP-00-51

KIRCHHOFF-INSTITUT FÜR PHYSIK

## Wissenschaftliche Arbeit

# Untersuchung eines mechanischen Oszillators zur Demonstration der Schwingungsgleichung

Helge Hauptfleisch

13. Juni bis 13. Dezember 2000



Verfaßt

am Kirchhoff-Institut

für Physik

der Universität Heidelberg

**Titelbild**: Dreidimensionale Phasenbahn einer periodischen, erzwungenen Schwingung. Erklärung siehe Abschnitt 4.5.

## Inhaltsverzeichnis

1	$\operatorname{Einl}$	eitung	4				
<b>2</b>	Funktionsweise des Pendels						
	2.1	Mechanischer Aufbau	7				
	2.2	Elektronische Schnittstelle	10				
	2.3	Regelkreis	11				
		2.3.1 Regelparameter	12				
		2.3.2 Realisierung als Computerprogramm	15				
3	Bedienung der Software						
	3.1	Das Programm pendeldemo	17				
	3.2	Das Programm resonanz	20				
	3.3	Das Programm chaos	22				
	3.4	Das Programm motorprüfstand	22				
4	Den	Demonstration der Schwingungsgleichung 2					
	4.1	Diskussion der Reibungsterme	25				
	4.2	Endlosschwingung	28				
	4.3	Mond- und Jupiterpendel	30				
	4.4	Resonanzverhalten	30				
	4.5	Von der Periodizität ins Chaos	35				
5	Schlußbetrachtung		40				
A Anhang		lang	42				
	A.1	Sourcecode	42				
		A.1.1 Regelschleife	42				
		A.1.2 Simulation des chaotischen Pendelverhaltens $\ldots$ $\ldots$ $\ldots$	47				
	A.2	Danksagung	49				
	A.3	Literaturverweise	50				
	A.4	Erklärung	51				

## 1 Einleitung

In dieser Arbeit wird ein mechanischer Oszillator vorgestellt, der dazu geeignet ist, die Schwingungsgleichung bis zum chaotischen Grenzfall zu demonstrieren. Es handelt sich hierbei um eine Weiterentwicklung des Oszillators, welcher von Herrn M.Öhmig im Rahmen der Wissenschaftlichen Arbeit [1] gebaut wurde. Die Hauptaufgabenstellung von Herrn Öhmig bestand in der technischen Realisierung eines physikalischen Pendels, auf das unabhängig von der jeweiligen Pendelposition ein sinusförmiges Drehmoment übertragen werden kann.

Die einzelnen Komponenten des Pendels erforderten noch umfangreiche Abstimmungen. So ist das System in einem ersten Schritt optimiert worden. Dazu waren sowohl Änderungen am mechanischen Aufbau, als auch eine grundlegende Überarbeitung der Steuerung durch den Computer notwendig. Die Funktionsweise der modifizierten Hardware und deren Ansteuerung ist Thema von Kapitel 2.

Zudem ist eine Software erstellt worden, die eine Bedienung des Pendels von einer grafischen Benutzeroberfläche aus ermöglicht und die über verschiedene Möglichkeiten der Datenaufnahme und der grafischen Darstellung von Meßergebnissen verfügt. Eine Anleitung zur Bedienung der Programme befindet sich in Kapitel 3, die zugehörigen Quelltexte stehen auszugsweise im Anhang unter A.1.1.

Kapitel 4 enthält eine phänomenologische Diskussion der Schwingungsgleichung anhand von Grafiken, die mit Hilfe des Oszillators aufgenommen wurden. In Kapitel 5 folgt eine kritische Betrachtung der Ergebnisse mit einem Ausblick auf die mögliche Weiterentwicklung des Oszillators.

## 2 Funktionsweise des Pendels

Der Oszillator besteht aus einem schwingfähigen Pendelkörper, einem Schrittmotor, einer Drehfeder, einem Drehgeber und einem elektronischen Interface. Dieses enthält neben den Spannungsversorgungen für Schrittmotor und Schrittmotortreiberplatine die Schnittstelle für den Computer, auf dem eine Software zum Auslesen der Pendelposition, Regeln der Schrittmotorposition und Ausgabe der gewünschten Daten implementiert ist.

Der Pendelkörper ist über eine horizontale Achse mit dem Drehgeber verbunden, dessen Position vom Computer ausgelesen werden kann. Der Drehgeber gibt dabei



Abbildung 1: Aufbau des mechanischen Oszillators. Durch den Zeiger wird die Stellung der Motorachse sichtbar.

einen Wert aus, den die Software in den Winkel  $\varphi$  zwischen Nullage und der momentanen Stellung des Pendels umrechnet, vergleiche Abbildung 2. Der Pendelkörper ist außerdem mit einer Drehfeder verbunden, deren relative Lage zur Pendelachse vom Schrittmotor gesteuert wird. Die Motorposition kann nicht ausgelesen werden, sondern errechnet sich aus dem Vorwert durch Addition bzw. Subtraktion des Schrittwinkels.

Diese Anordnung erlaubt es, beliebige Drehmomente  $T^*$  auf den Pendelkörper zu übertragen, wobei  $T^*$  sowohl von den beiden direkten Meßgrößen Pendelposition  $\varphi$  und der Zeit t abhängen kann, als auch von Rechengrößen wie der Pendelgeschwindigkeit  $\dot{\varphi}^1$ . Bei bekanntem Richtmoment  $D^o$  der Drehfeder wird der Motor gerade so geregelt, daß der Winkel  $\psi$  zwischen Pendel- und Motorachse Gleichung 1 erfüllt.

$$\psi\left(\varphi,\dot{\varphi},t\right) = -\frac{T^*\left(\varphi,\dot{\varphi},t\right)}{D^o} \tag{1}$$

Dazu wird in bestimmten Zeitintervallen die Position des Pendels  $\varphi$  ausgelesen,  $\dot{\varphi}$  berechnet und der Winkel  $\tilde{\psi}(\varphi, \dot{\varphi}, t)$  relativ zur Motorachse bestimmt. Ist die Abweichung vom geforderten Wert  $\psi(\varphi, \dot{\varphi}, t)$  aus Gleichung 1 größer als ein durch das Auflösungsvermögen des Schrittmotors gegebener Toleranzbereich, wird die Stellung der Motorachse korrigiert.

<sup>&</sup>lt;sup>1</sup>Es gilt  $\dot{a} := \frac{da}{dt}$ .



Abbildung 2: Definition des Koordinatensystems. Die z-Achse verläuft parallel zur Drehachse und zeigt aus der Papierebenen heraus. Der Drehgeber befindet sich bei dieser Darstellung auf der negativen, die Drehfeder und der Schrittmotor auf der positiven z-Achse.

#### 2.1 Mechanischer Aufbau

Der Pendelkörper besteht aus einem 20 cm langen zylinderförmigen Aluminiumstab mit 12 mm Durchmesser. Er ist im Abstand von 5 cm vom oberen Ende aus drehbar auf einer horizontalen Achse gelagert. Am unteren Ende kann ein Gewichteteller mit maximal sechs Edelstahlscheiben aufgeschraubt werden. Somit nimmt das Trägheitsmoment  $\theta$  die in Tabelle 1 aufgelisteten Werte an.  $\theta$  berechnet sich nach dem Steinerschen Satz als Summe aus dem Trägheitsmoment  $\theta_0$  bezüglich der zur Schwingungsachse parallelen Achse durch den Schwerpunkt und dem Trägheitsmoment der im Schwerpunkt vereinigten Gesamtmasse m bezüglich der ursprünglichen Achse ([3]).

$$\theta = \theta_0 + mr^2 \tag{2}$$

r ist der Abstand zwischen Drehachse und Schwerpunkt. Bei den Werten in Tabelle 1 sind zusätzlich die Trägheitsmomente der horizontalen Verbindungsachse und der Lagerscheiben berücksichtigt. Durch den Einfluß der Drehfeder und deren Befestigungsteile kann sich der Wert des Trägheitsmoments um maximal 3 % erhöhen. Diese Unsicherheit dominiert die übrigen Meßungenauigkeiten.

Nach dem Drehimpulssatz für Vielteilchensysteme verhält sich ein physikalisches Pendel so, als sei die gesamte Masse m im Abstand r des Schwerpunktes von der Drehachse vereint, vergleiche Abbildung 2. Für das maximal rücktreibende Moment

	m[g]	r[cm]	$ heta[gm^2]$	$T_{r,max}[mNm]$
Pendelstab	69,0	3,9	0,36	26,4
mit 1 Scheibe	110,9	8,15	1,56	88,6
mit 2 Scheiben	146,4	10,1	2,30	145,0
mit 3 Scheiben	181,8	11,2	3,28	199,7
mit 4 Scheiben	217,3	12,2	4,33	260,0
mit 5 Scheiben	252,7	13,0	5,42	322,2
mit 6 Scheiben	288,2	13,5	6,61	381,5

**Tabelle 1:** Konstanten des Pendelörpers: Gesamtmasse m, Abstand Drehachse - Schwerpunkt r, Trägheitsmoment  $\theta$  und maximal rücktreibendes Moment  $T_{r,max}$ . Maximaler Fehler 3 %.

folgt Gleichung 3, g ist die Fallbeschleunigung der Erde.

$$T_{r,max} = m \cdot g \cdot r \tag{3}$$

Für allgemeine Winkel  $\varphi$  zwischen Nullage und Momentanposition des Pendels gilt Gleichung 4.

$$T_r\left(\varphi\right) = -T_{r,max} \cdot \sin\left(\varphi\right) \tag{4}$$

Zur Bestimmung der Charakteristik der Drehfeder wurde der Winkel  $\psi$  zwischen Pendel- und Motorachse gemessen, der erforderlich ist, um das Pendel in der Lage  $\varphi$ zu halten. Das Drehmoment folgt dann aus Tabelle 1 und Gleichung 4. Bei positiven Winkeln ist das Richtmoment systemmatisch größer als bei negativen, vergleiche Abbildung 3. Die Software korrigiert diese Abweichung vom Hookschen Gesetz auf  $D^o = 0,251 \frac{Nm}{rad} \pm 5 \%$ .

Ein Schrittmotor ist ein Elektromotor, dessen Achse nur diskrete Positionen annehmen kann. Das Auflösungsvermögen des verwendeten Schrittmotors beträgt 400 Schritte pro Umdrehung, das entspricht  $0,9^{\circ}$  bzw.  $0,005\pi$ . Nach Herstellerangaben ist die Genauigkeit der Achsstellung 5% und das maximale Haltemoment 0,494 Nm. Im Belastungstest werden jedoch nur  $380 Nm \pm 5\%$  gemessen.

Der Drehgeber teilt eine volle Umdrehung in 8192 Schritte, das Auflösungsvermögen ist demnach mit  $0,00024 \pi$  etwa zwanzigmal größer als beim Schrittmotor, was beim Vergleich der beiden Werte zu beachten ist.



Abbildung 3: Charakteristik der Drehfeder mit Regressionsgeraden. Die unterschiedlichen Steigungen für positive und negative Winkel sind bei dieser Darstellung kaum erkennbar.



Abbildung 4: Schaltplan der Schnittstelle (aus [1]). Der Stecker S1 ist mit dem Computer verbunden, S2 mit der Schrittmotortreiberplatine. An S3 wird der Drehgeber ausgelesen und über S4 werden die Befehle der Schrittmotortreiberplatine an die Motorspulen weitergeleitet.

#### 2.2 Elektronische Schnittstelle

Die Elektronik zur Übermittelung der Daten zwischen Computer, Schrittmotor und Drehgeber befindet sich in einem Interface, dessen Schaltung ist in Abbildung 4 gezeigt.

Der Stecker S1 ist über ein spezielles Kabel mit der digitalen I/O-Karte DIO-24<sup>2</sup> des Computers verbunden, welche über die drei 8 Bit-Ports JA, JB und JC verfügt. JA ist als Ausgabeport für den Computer definiert, JB und JC als Eingabeport. JCwird allerdings nicht benötigt. In Tabelle 2 sind die Ports und Pinnummern der Signale zur Hardwareansteuerung aufgelistet. Logisch 1 bedeutet, daß an dem entsprechenden Pin eine Spannung von 5 V anliegt, bei logisch 0 liegt keine Spannung an (TTL-Logik). Vergleichspotential ist *Ground* auf Pin 49. Der Zugriff auf die Ports erfolgt byteweise. Eine Veränderung der Belegung einzelner Pins wird durch die Verknüpfung der Signale mit logischen Operatoren erreicht.

Der Drehgeber gibt die Position seiner Achse seriell auf Port JB über die Datenleitungen  $DG\_DATAM$  und  $DG\_DATABM$  aus. Dazu werden dreizehn Taktimpulse über  $DG\_TAKTM$  und  $DG\_TAKTBM$  an JA angelegt. Die Karte erlaubt die Drehgeberauslese in etwa 50  $\mu s$ . Nach frühestens 20  $\mu s$  ist eine Positionsabfrage erneut möglich.

<sup>&</sup>lt;sup>2</sup>DAQ $Card^{TM}$ -DIO-24, National Instruments

Signal	Dualdarstellung	Eigenschaft	Port	Pinnummer
DG_DATABM	00000010	inverser Datenpuls	JB	29
DG_DATAM	00000001	Datenpuls	JB	31
M_PRESETM	10000000	Inbetriebnahme des Motors	JA	33
M_DIRECTM	01000000	$\operatorname{Drehrichtung}$	JA	35
M_CLOCKM	00100000	$\operatorname{Schrittpuls}$	JA	37
$M\_STEPM$	00010000	Voll-\Halbschrittmodus	JA	39
DG_TAKTBM	00000010	inverser Taktpuls	JA	45
$DG_{-}TAKTM$	00000001	Taktpuls	JA	47

**Tabelle 2:** Eigenschaft, Port und Pinnummer der Signale an Stecker S1.

Das Ansteuern des Motors erfolgt über  $M\_PRESETM$ ,  $M\_DIRECTM$ ,  $M\_CLOCKM$ und  $M\_STEPM$ . Bei jeder negativen Flanke von  $M\_CLOCKM$  dreht sich die Motorachse um einen Schritt in die eingestellte Richtung im eingestellten Modus. Die eintreffenden Pulse werden vom Treiberbaustein 7407N verstärkt und von der Schrittmotortreiberplatine in Befehle für die Spulen des Motors umgewandelt.

## 2.3 Regelkreis

"Als Regelung wird jeder Vorgang in einem abgegrenzten System bezeichnet, bei dem eine oder mehrere physikalische, technische (insbesondere verfahrenstechnische) oder andere Größen, die Regelgrößen x, fortlaufend von einer Meßeinrichtung erfaßt und durch Vergleich ihrer jeweiligen Augenblickswerte, den Ist-Werten  $x_i$ , mit den vorgegebenen Soll-Werten  $x_s$  bestimmter vorgegebener Führungsgrößen w auf diese Werte gebracht (im Sinne einer Angleichung) und dann auf ihnen gehalten werden. Im Gegensatz zur Steuerung von Größen bzw. Vorgängen wird bei einer Regelung das Ergebnis der Verstellung ihrer Größen durch fortwährende Messung kontrolliert und, wenn erforderlich, korrigiert."([2])

Sollen Drehmomente  $T^*(\varphi, \dot{\varphi}, t)$  auf das Pendel übertragen werden, ist die Regelgröße der Winkel  $\tilde{\psi}$  zwischen Pendel- und Motorposition. Die Führungsgröße  $\psi$  ist durch Gleichung 1 gegeben. Abbildung 5*b* zeigt die Orientierung von  $\psi$ . Die Regelabweichung y(t) ist die Differenz aus den momentanen Soll- und Ist-Werten.

$$y(t) = \psi\left(\varphi, \dot{\varphi}, t\right) - \tilde{\psi}\left(\varphi, \dot{\varphi}, t\right)$$
(5)



**Abbildung 5:** a) Definition des Winkels  $\varphi$  zwischen Nullage und Momentanposition des Pendels b) Orientierung des Winkels  $\psi$  zwischen Pendel- und Motorachse

Die Durchlaufgeschwindigkeit k der Regelschleife ist eine Funktion von y(t). Diese wird vom Benutzer durch die Wahl der Regelparameter festgelegt.

#### 2.3.1 Regelparameter

Die zeitliche Steuerung der Regelschleife erfolgt über die Regelparameter minimale Durchlaufgeschwindigkeit  $k_{min}$  und Steigung m, vergleiche Abbildung 6. Die Regelgeschwindigkeit k ist im Bereich von  $100 Hz \leq k \leq 1300 Hz$  durch den Betrag der Regelabweichung definiert.

$$k = m \cdot (|y| - 0,005) \pi + k_{min} \tag{6}$$

Für *m* können Werte von Null bis  $200 \frac{kHz}{\pi}$  gewählt werden, der Wertebereich von  $k_{min}$  ist 100 Hz bis 1300 Hz. Ist m = 0, wird die Regelschleife in äquidistanten Zeitintervallen durchlaufen. Andernfalls geht die Regelgeschwindigkeit in einem bestimmten Bereich linear mit der Regelabweichung.

Jede Regelabweichung überträgt nach Gleichung 1 ein Drehmoment, das die Pendelbewegung beeinflußt, vergleiche Abbildung 7. Je schneller die Regelschleife durchlaufen wird, desto schneller werden Regelabweichungen erkannt und behoben.

• Je größer  $k_{min}$  und m, desto weniger beeinflußt die Regelung die Pendelbewegung.

Bei Soll-Werten, die explizit von der Pendelgeschwindigkeit  $\dot{\varphi}$  abhängen, gibt es bei großen Werten von  $k_{min}$  allerdings ein Digitalisierungsproblem. Da  $\dot{\varphi}$  diskret durch



Abbildung 6: Parameter des Regelkreises. Der Toleranzbereich von  $0,005 \pi$  ist ebenso wie die maximale Regelgeschwindigkeit von 1300 Hz durch den Schrittmotor begrenzt.

den Vergleich zweier aufeinanderfolgender Pendelpositionen berechnet wird, ist die Genauigkeit der Geschwindigkeitsmessung durch das Auflösungsvermögen des Drehgebers begrenzt ( $\delta_{DG} = \frac{2\pi}{8192}$ ). Die kleinste meßbare Winkelgeschwindigkeit  $\dot{\varphi}_{step}$  ist proportional zu  $\delta_{DG}$ .

$$\dot{\varphi}_{step} = \delta_{DG} \cdot k_{min} \tag{7}$$

Bei kleinen Werten von  $k_{min}$  ist die Auflösung der Geschwindigkeitsmessung größer. Die Geschwindigkeits-Zeit-Kurve und der Soll-Wert werden somit glatter.

• Je größer die  $\dot{\varphi}$ -Abhängigkeit des Soll-Wertes, desto kleiner sollten  $k_{min}$  und m gewählt werden.

Der Einfluß der Regelung auf die Pendelbewegung entspricht in guter Näherung einem Stokesschen Reibungsterm, der für gegebene Regelparameter gemessen werden kann. Dies erfolgt in Abschnitt 4.1. In den meisten Fällen ist es vorteilhaft, niedrige Werte für  $k_{min}$  und m zu verwenden und den zusätzlichen Reibungsterm über eine entsprechende Wahl des Soll-Wertes auszugleichen.

Als dritter Regelparameter kann optional eine maximale Motorbeschleunigung G im Bereich von 1000 Motorschritten/ $s^2$  bis 1,69 · 10<sup>6</sup> Motorschritten/ $s^2$  aktiviert werden. Außerhalb des Toleranzbereiches entspricht die Regelgeschwindigkeit gerade der



**Abbildung 7:** Messung der Pendelposition  $\varphi$  und des Winkels  $\tilde{\psi}$  zwischen Pendel- und Motorachse beim Sollwert  $\psi = 0$ . In a) wurde der Motor abmontiert, bei b) beeinflußt die Regelabweichung die Schwingung.

Motorgeschwindigkeit. Beim Übergang der Regelgeschwindigkeit  $k_v$  zu k, erfährt die Motorachse dem Betrage nach die Winkelbeschleunigung  $\dot{\omega}_{mot}$ .

$$\dot{\omega}_{mot} = \frac{\mid k_v - k \mid}{\frac{1}{k_v}} \tag{8}$$

Diese ist einem Drehmoment  $T_{mot}$  proportional, das zusätzlich zu  $T^*(\varphi, \dot{\varphi}, t)$  auf die Motorachse wirkt. Wird das maximale Haltemoment des Schrittmotors überschritten, rutscht die Motorachse durch. Dies kann kurzfristig durch eine Beschleunigungsbegrenzung verhindert werden. Bei maximal zulässiger Beschleunigung G ergibt sich der Maximalwert für k nach Gleichung 9.

$$k = \frac{G}{k_v} + k_v \qquad \text{bzw.} \qquad k = k_v - \frac{G}{k_v} \tag{9}$$

Die Beschleunigungsbegrenzung muß nur dann aktiviert werden, wenn das System durch die Wahl des Soll-Wertes und die Anzahl der aufgelegten Gewichtscheiben an seine Grenzen stößt. Der Einfluß auf die Regelung ist äußerst komplex. Statt die Beschleunigungsbegrenzung zu aktivieren ist es in der Regel sinnvoller, die Anzahl der Gewichtscheiben zu reduzieren.



 $\Delta t = 0.1 \text{ms}$ 

Abbildung 8: Oszillatorbild der Regelschleife

## 2.3.2 Realisierung als Computerprogramm

In Abbildung 8 ist der zeitliche Ablauf der Regelschleife als Oszillatorbild dargestellt.

- A Bestimmung der Zeit t mit dem Taktgeber des Computerprozessors. Bei Beginn der Regelschleife ist t = 0.
- ${\bf B}$ Auslesen des Drehgebers und Berechnen der Pendelposition. Die Drehgeberauslese darf nicht durch betriebssysteminterne Interrupts unterbrochen werden, vergleiche PunktF.
- C Ist der Betrag der im vorangegangenen Schleifendurchlauf gemessenen Regelabweichung größer als der Toleranzbereich, wird eine fallende Flanke auf  $M\_CLOCKM$ erzeugt und damit der Motor um einen Schritt in die auf  $M\_DIRECTM$  eingestellte Richtung bewegt. Im ersten Durchlauf ist die Regelabweichung Null.
- **D** Berechnen der Pendelgeschwindigkeit  $\dot{\varphi}$  und der gewünschten Daten aus den beiden direkten Meßgrößen Pendelposition  $\varphi$  und Zeit t.
- ${\bf E}\,$  Aufnahme der Werte in den Arbeitsspeicher.
- F Aktualisierung der Benutzeroberfläche. Jede Zehntelsekunde werden die anstehenden Prozesse für maximal eine Millisekunde abgearbeitet, angefangene Prozesse können nicht unterbrochen werden. Nur in dieser Zeit sind betriebssysteminterne Interrupts zugelassen, während der Windows '98 den Status der Tastatur und der Maus abfragt und den Bilschirm aktualisiert.



**Abbildung 9:** Meßergebnis der Regelcharakteristik bei  $k_{min} = 100$  und  $m = 200 \frac{kHz}{\pi}$ . Die Ausreißer sind auf Aktualisierungsprozesse zurückzuführen.

- G Berechnen der neuen Regelabweichung.
- **H** Je nach Vorzeichen der Regelabweichung wird die Richtung der nächsten Motorbewegung mit dem Signal *M\_DIRECTM* eingestellt und die neue Motorposition berechnet.
- I Je nach Größe des Betrages der Regelabweichung und Wahl der Regelparameter wird die Restwartzeit dynamisch bestimmt und eine entsprechende Warteschleife gestartet. Die Zeit pro Schleifendurchlauf ist gerade der Kehrwert aus Gleichung 6, vergleiche Abbildung 9.

## 3 Bedienung der Software

Zur Demonstration der Schwingungsgleichung stehen drei Programme zur Verfügung. Die Quelltexte mit erläuternden Kommentaren zur Programmierung befinden sich im Anhang unter A.1.1. Die Anwendungsgebiete sind folgendermaßen aufgeteilt:

pendeldemo: Allgemeine Demonstrationen

resonanz: Demonstration des Resonanzverhaltens

chaos: Demonstration des chaotischen Verhaltens

Vor dem Einschalten der Spannungsversorgung wird die Motorachse so gedreht, daß die Drehfeder in der Ruhelage des Pendels möglichst entspannt ist. Bei Programmstart wird diese Lage als Nullage des Pendels und des Motors gespeichert. Die oberen drei bzw. zwei Schieberegler der Benutzeroberflächen repräsentieren die in 2.3.1 diskutierten Regelparameter m,  $k_{min}$  und G. Nach Beendigung einer Messung erscheinen je nach Programm ein oder mehrere Grafikfenster. Alle aufgenommenen Meßdaten werden außerdem tabellarisch in das Verzeichnis .\messwerte als Datenfile gespeichert und können so mit einem Grafikprogramm weitergehend bearbeitet werden. Zur Steuerung des Programmablaufs dienen die Knöpfe:

- **Start** Startet einen Regelkreis mit den gewählten Parametern, die Systemzeit wird zu Beginn auf Null gesetzt. Eine Veränderung sämtlicher Schiebereglerwerte ist weiterhin möglich.
- Stop Stoppt die Regelschleife. Die Motorachse ist in der momentanen Position fixiert. Je nach Programm erscheinen ein oder mehrere Grafikfenster.
- **Pendel auf Null fahren** Das Pendel wird in die Position gefahren, die bei Programmstart als Nullage gespeichert wurde. Anschließend wird die Motorposition Null gesetzt. So wird gewährleistet, daß nach einem eventuellen Durchrutschen der Motorachse deren richtige Position wieder bekannt ist. Diese Taste kann auch direkt zur Beendigung der Regelschleife verwendet werden. Während die Funktion läuft, werden andere Befehle ignoriert.
- **Grafik löschen** Die Kurven aller Plotfenster werden gelöscht. Wird diese Funktion nicht aufgerufen, überlagern sich die Kurven bei einer neuen Messung.
- Schließen Alle Plotfenster und das Hauptfenster werden geschlossen, das Programm ist beendet. Die Fixierung der Motorachse wird durch Ausschalten der Spannungsversorgung aufgehoben.

## 3.1 Das Programm pendeldemo

Nach dem Starten von "pendeldemo" erscheint die in Abbildung 10 gezeigte Benutzeroberfläche. Wird das Kontrollkästchen "Aufnahme von Meßwerten" aktiviert, erscheint nach Betätigung der *Stop*-Taste oder, wenn die angegebene Anzahl der Meßwerte erreicht wurde, ein Weg-Zeit-Diagramm ähnlich Abbildung 11 *a*. Weitere Diagramme zu



Abbildung 10: Benutzeroberfläche von "pendeldemo".

demselben Versuch erhält man über die entsprechenden Knöpfe des umrandeten Feldes "Plotfenster öffnen", vergleiche Abbildung 11 a bis e. Der eingestellte Wert der Anzahl der Meßwerte wird beim Starten der Regelschleife gespeichert und kann nachträglich nur erniedrigt werden.



Abbildung 11: Grafikfenster und Plots der zusätzlich auf Festplatte gespeicherten Daten von "pendeldemo", hier eine sinusförmigen Anregung in der Einschwingphase. Bei der auf Festplatte gespeicherten Geschwindigkeit in g wird aus jeweils fünf Werten von b der Mittelwert gebildet, dadurch wird die Kurve glatter. Die externe Phase in h wird in Abschnitt 4.5 definiert.

Datenfile	x-Wert, y-Wert, z-Wert der Meßpunkte	grafische Darstellung
weg_zeit.dat	${\rm Zeit,  Pendel position, } \emptyset$	Abbildung $11 a$
phasenraum.dat	${\rm Pendelposition,  Pendelgeschwindigkeit,  \emptyset}$	Abbildung $11c$
$motorpos\_zeit.dat$	Zeit, Motorposition, $\emptyset$	Abbildung $11 d$
winkel.dat	$\text{Zeit}, \tilde{\psi}, \emptyset$	Abbildung $11e$
${ m amplitude.dat}$	${\rm Zeit, \ Amplitude, \ } \emptyset$	Abbildung 11 $f$
$geschw\_zeit.dat$	${\rm Zeit,  Pendelgeschwindigkeit, } \emptyset$	Abbildung $11g$
$pos\_winkel.dat$	Zeit, Pendelposition & $\tilde{\psi}$ , Ø	Abbildung 7
${ m mspeed.dat}$	${\rm Regelabweichung, Regelgeschwindigkeit, } \emptyset$	Abbildung 9 $a$
wartez eit.dat	Regelabweichung, Zeit pro Regeldurchlauf, $\emptyset$	Abbildung 9 $b$
3D.dat	$\phi$ , Pendelposition, Pendelgeschwindigkeit	Abbildung $11 h$

**Tabelle 3:** Verzeichnisse der gespeicherten Meßpunkte von "pendeldemo". Die externe Phase  $\phi$  wird in Abschnitt 4.5 definiert.

## 3.2 Das Programm resonanz

Zur Untersuchung des Resonanzverhaltens wird ein sinusförmiges Drehmoment auf das Pendel übertragen.

$$T(t) = T_0^* \cdot \sin(\omega t) \tag{10}$$

Die Kreisfrequenz  $\omega$  durchläuft schrittweise einen vorgegebenen Wertebereich. Startund Stopfrequenz werden ebenso wie der Wert der Erhöhung von 0, 1 Hz bis 1 Hz über einen Schieberegler eingestellt. Der aktuelle Wert der Anregungsfrequenz wird jeweils vom vorletzten Schieberegler angenommen.

Das Programm mißt nach dem Einschwingvorgang zu jedem Wert von  $\omega$  die Amplitude des Pendels sowie die Phasenverschiebung zwischen anregendem Moment und Auslenkung. Die Erhöhung der Frequenz erfolgt nach der Zeit  $t_{step}(\omega)$ , die Meßwerte werden jeweils in der letzten Anregungsperiode vor der Frequenzerhöhung aufgenommen.

$$t_{step}(\omega) = C \cdot \operatorname{ceil}(\omega) \cdot \frac{2\pi}{\omega}$$
(11)

Die Funktion ceil( $\omega$ ) ordnet jedem  $\omega$  den kleinsten ganzzahligen Wert zu, der nicht kleiner ist als  $\omega$ . Die Konstante C ist der Wert des Kästchens "Dauer des Einschwingvorgangs". Für C = 1 folgt  $t(\omega) \sim 10 s$ , bei C = 4 wird die Frequenz etwa alle 30 Sekunden erhöht. Ein mögliches Ergebnis zeigt Abbildung 13.



Abbildung 12: Benutzeroberfläche von "resonanz".



Abbildung 13: Grafikfenster von "resonanz". Im Bereich zwischen 6 Hz und 7Hz betrug die Frequenzerhöhung 0.2 Hz, ansonsten 0.5 Hz. Die zugehörigen Tabellen finden sich in .\resonanz.dat bzw. .\phase.dat.



Abbildung 14: Benutzeroberfläche von "chaos".

### 3.3 Das Programm chaos

Zur Untersuchung des chaotischen Verhaltens des Pendels wird ein sinusförmiges Drehmoment analog Gleichung 10 übertragen. Das Programm gibt sogenannte Poincaréplots aus. Bei dieser Form der Darstellung wird in äquidistanten Zeitintervallen  $\Delta t = \frac{2\pi}{\omega}$  je ein Meßpunkt im Phasenraum aufgenommen, vergleiche Abbildung 15. Die Aufnahme des ersten Meßpunktes erfolgt nach Beendigung des Einschwingvorganges, der hier in Perioden anzugeben ist, und nachdem das anregende Moment die im letzten Schieberegler eingestellte Phase durchlaufen hat. Die Tabelle der aufgenommenen Daten wird in .\chaos.dat gespeichert.

## 3.4 Das Programm motorprüfstand

Beim Betätigen des Startknopfes dreht sich der Motor mit der angegebenen Geschwindigkeit um die angegebene Schrittzahl. Die Richtung kann durch aktivieren bzw. deaktivieren des entsprechenden Kontrollkästchens variiert werden. Der zulässigen Maximalwert der Motorbeschleunigung wird über den zweiten Schieberegler eingestellt, vergleiche Abbildung 16.



Abbildung 15: Poincaréplot mit 1000 Meßpunkten.



Abbildung 16: Benutzeroberfläche von motorprüfstand

## 4 Demonstration der Schwingungsgleichung

Nach dem Drehimpulssatz entspricht "*die zeitliche Änderung des Gesamtdrehimpulses* ... *der Summe der äußeren Drehmomente*" ([3]). Bei der z-Komponente (vergleiche Abbildung 2) gilt dieser Zusammenhang für die Beträge.

$$\theta\ddot{\varphi} = \sum_{i=1}^{n} T_n^{(ex)} \tag{12}$$

Die äußeren Drehmomente  $T_n^{(ex)}$  setzen sich zusammen aus dem durch die Drehfeder übertragenen Moment  $T^*(\varphi, \dot{\varphi}, t)$  (vergleiche Gleichung 1), dem rücktreibenden Drehmoment  $T_r(\varphi)$  (vergleiche Gleichung 4) und Reibungstermen. Diese bestehen hauptsächlich aus einem geschwindigkeitsproportionalen<sup>3</sup> und einem konstanten<sup>4</sup> Anteil.

$$\theta \ddot{\varphi} = T^* \left( \varphi, \dot{\varphi}, t \right) - T_{r,max} \cdot \sin\left(\varphi\right) - b \dot{\varphi} - \operatorname{sgn}(\dot{\varphi})c \tag{13}$$

Für  $\dot{\varphi} \neq 0$  ist sgn $(\dot{\varphi})$  der Quotient  $\frac{\dot{\varphi}}{|\dot{\varphi}|}$ , für  $\dot{\varphi} = 0$  gilt sgn $(\dot{\varphi}) = 0$ .

Die Schieberegler der grafischen Benutzeroberfläche definieren  $T^*(\varphi, \dot{\varphi}, t)$  in der Form von Gleichung 14.

$$T^*\left(\varphi,\dot{\varphi},t\right) = T_0^* \cdot \sin\left(\omega t\right) + T_{r,max}^* \cdot \sin\left(\varphi\right) + b^*\dot{\varphi} + \operatorname{sgn}(\dot{\varphi})c^*$$
(14)

Die allgemeine Bewegungsgleichung für das Pendel ist somit Gleichung 15.

$$\theta\ddot{\varphi} = \left(T_{r,max}^* - T_{r,max}\right) \cdot \sin\left(\varphi\right) + T_0^* \cdot \sin\left(\omega t\right) + \left(b^* - b\right)\dot{\varphi} + \operatorname{sgn}(\dot{\varphi})\left(c^* - c\right)$$
(15)

Alle Parameter der Schwingungsgleichung können über Schieberegler künstlich vergrössert oder verkleinert werden. Die Wirbelstrombremse, die in [1] dazu gedacht war, die Stokesschen Reibungsterme zu verstärken, wurde abgebaut.

<sup>&</sup>lt;sup>3</sup>Luftwiderstand  $\rightarrow$  Stokesreibung

<sup>&</sup>lt;sup>4</sup>Rollreibung in den Lagern, Innenwiderstand des Drehgebers  $\rightarrow$  Coulombreibung



Abbildung 17: Pendelschwingung mit unterschiedlichen Reibungsarten: a) Reine Coulombreibung, die Einhüllende ist eine Gerade. b) Reine Stokesreibung, die Einhüllende ist eine Exponentialfunktion.

#### 4.1 Diskussion der Reibungsterme

Unter der Annahme  $\varphi \ll \frac{\pi}{2}$  (Kleinwinkelnäherung) und  $T_0^* = 0$  (kein anregendes Moment) vereinfacht sich die allgemeine Bewegungsgleichung 15 zu Gleichung 16.

$$\theta \ddot{\varphi} = -T_{r,max} \cdot \varphi + (b^* - b) \dot{\varphi} + \operatorname{sgn}(\dot{\varphi}) (c^* - c)$$
(16)

Die Pendelschwingung ist wesentlich durch die Art der vorherrschenden Reibungsterme charakterisiert, vergleiche Abbildung 17. Bei reiner Stokesreibung ist die Einhüllende eine Exponentialfunktion, bei reiner Coulombreibung eine Gerade<sup>5</sup>. Eine Eliminierung der konstanten Reibungsterme wird mit  $c^* = 7, 1 \pm 0, 4 \, mNm$  erreicht.

$$\theta \ddot{\varphi} = -T_{r,max} \cdot \varphi + (b^* - b) \dot{\varphi} \tag{17}$$

Bei der Lösung dieser linearen Differentialgleichung zweiter Ordnung unterscheidet man drei Fälle ([5]):

1.  $b - b^* < \sqrt{\theta \cdot T_{r,max}}$  (schwache Dämpfung, Schwingfall): Mit den Anfangsbedingungen  $\varphi(t=0) = \varphi_0$  und  $\dot{\varphi}(t=0) = 0$  ist Gleichung 18 Lösung der Bewegungsgleichung:

$$\varphi(t) = \varphi_0 \cdot e^{-\frac{b-b^*}{2\theta}t} \cdot \cos(\varpi t) \quad \text{mit} \quad \varpi = \sqrt{\frac{T_{r,max}}{\theta} - \left(\frac{b-b^*}{2\theta}\right)^2} \quad (18)$$

 $<sup>{}^{5}</sup>$ Wird der Motor abgebaut, dominiert der konstante Innenwiderstand des Drehgebers die Schwingung, vergleiche Abbildung 7 a.



Abbildung 18: Halblogarithmische Darstellung der Einhüllenden bei  $c^* = 7, 1 \, mNm$ . Die Regelung induziert einen Stokesschen Reibungsterm, dessen Größe von der Wahl der Regelparameter abhängt. Diese Eichung gilt für Soll-Werte, die nicht explizit von der Pendelpostion  $\varphi$  abhängen.

- 2.  $b b^* = \sqrt{\theta \cdot T_{r,max}}$  (mittlere Dämpfung, aperiodischer Grenzfall): Je nach Anfangsbedingungen gibt es beim aperiodischen Grenzfall maximal noch einen Nulldurchgang.
- 3.  $b b^* > \sqrt{\theta \cdot T_{r,max}}$  (starke Dämpfung, Kriechfall): Mit den Anfangsbedingungen aus 1. ist kein Nulldurchgang mehr möglich.

Im Schwingfall kann der Stokessche Reibungsterm *b* über die Steigung der Einhüllenden bei halblogarithmischer Darstellung bestimmt werden. Dieser hängt von der Wahl der Regelparameter ab, vergleiche Abbildung 18. Abbildung 19 zeigt den Übergang vom Schwingfall zum Kriechfall.



**Abbildung 19:** a) Schwingfall, b) aperiodischer Grenzfall, c) Kriechfall. Beim aperiodischen Grenzfall wird das System schneller gedämpft als beim eigentlichen Kriechfall (vgl. [3]).



Abbildung 20: Die Schwingungsdauer T hängt bei großen Auslenkungen von der Amplitude ab. Die Meßwerte stimmen bis auf 1,5 % mit dem Rechenwert aus Gleichung 20 überein.

## 4.2 Endlosschwingung

Bei bekannten Werten von b und c kann eine Endlosschwingung erzeugt werden. Diese ist im Falle der Kleinwinkelnäherung harmonisch, vergleiche Abbildung 21. Eine harmonischen Schwingung stellt im Phasenraum eine Ellipse dar, deren Halbachsen **a** und **b** nach [4] von der Gesamtenergie E des Systems abhängen.

$$E = \frac{1}{2}\theta\dot{\varphi}^2 + \frac{1}{2}T_{r,max}\varphi^2 \qquad \mathbf{a} = \sqrt{\frac{2E}{\theta}} \qquad \mathbf{b} = \sqrt{\frac{2E}{T_{r,max}}} \tag{19}$$

Mit den Werten von  $\varphi$  und  $\dot{\varphi}$  aus Abbildung 21 *a* und *b* folgt  $\mathbf{a} = 2,04 \frac{\pi}{s}$  bzw.  $\mathbf{b} = 0,26 \pi$  für die kleine bzw. große Halbachse, was in Abbildung 21 *c* bestätigt ist.

Die Frequenz der gedämpften Schwingung  $\varpi$  aus Gleichung 18 ist strenggenommen amplitudenabhängig, da das Pendel selbst bei kleinen Auslenkungen nur näherungsweise ein harmonischer Oszillator ist. Mit einer reibungsfreien Endlosschwingung kann die Abhängigkeit der Schwingungsdauer **T** von der Amplitude  $\varphi_0$  untersucht werden. Löst man Gleichung 19 nach  $d\varphi$  auf und integriert von  $-\varphi_0$  bis  $\varphi_0$ , erhält man ein elliptisches Integral erster Ordnung. Die Lösung kann als Reihenentwicklung angegeben werden, vergleiche [6] und [7]:

$$\mathbf{T}(\varphi_0) = 2\pi \sqrt{\frac{\theta}{T_{r,max}}} \cdot \left(1 + \sum_{i=1}^{\infty} \left(\prod_{j=1}^i \frac{2j-1}{2j}\right)^2 \sin^{2i}\left(\frac{\varphi_0}{2}\right)\right)$$
(20)

In Abbildung 20 wurde die Schwingungsdauer in Abhängigkeit der Amplitude bei einer Endlosschwingung gemessen und nach Gleichung 20 berechnet.



Abbildung 21: Harmonische Schwingung a) Weg-Zeit-Diagramm b) Geschwindigkeits-Zeit-Diagramm c) Phasenraum. Die Phasenbahn einer harmonischen Schwingung ist eine Ellipse.

### 4.3 Mond- und Jupiterpendel

Wird dem Wert des rücktreibenden Momentes  $T_{r,max}^*$  ein Wert ungleich Null zugewiesen, wirkt sich dies auf die Pendelbewegung wie eine künstliche Veränderung der Erdbeschleunigung g aus. Aus dem Newtonschen Gravitationsgesetz folgt für die Fallbeschleunigung des Mondes  $g_{Mond} \approx \frac{1}{6} \cdot g$ , auf dem Jupiter<sup>6</sup> gilt  $g_{Jupiter} \approx 2, 5 \cdot g$ . Um das Verhalten eines physikalischen Pendels bei der Fallbeschleunigung des Mondes bzw. Jupiters zu untersuchen, muß  $T_{r,max}^*$  die Werte aus Gleichung 21 annehmen.

$$T_{r,max}^* = \frac{5}{6} \cdot T_{r,max}$$
 bzw.  $T_{r,max}^* = -1, 5 \cdot T_{r,max}$  (21)

Für den Quotienten der Schwingungsdauern bei der Amplitude  $\varphi_0$  gilt Gleichung 22.

$$\frac{\mathbf{T}^{Erde}(\varphi_0)}{\mathbf{T}^{Mond}(\varphi_0)} = \sqrt{\frac{1}{6}} \qquad \text{bzw.} \qquad \frac{\mathbf{T}^{Erde}(\varphi_0)}{\mathbf{T}^{Jupiter}(\varphi_0)} = \sqrt{2,5}$$
(22)

Abbildung 22 bestätigt diesen Sachverhalt.

#### 4.4 Resonanzverhalten

Ausgangspunkt bei der Untersuchung des Resonanzverhaltens ist die Grundgleichung der erzwungenen Schwingung.

$$\theta \ddot{\varphi} + (b - b^*) \dot{\varphi} + T_{r,max} \cdot \varphi = T_0^* \cdot \sin(\omega t)$$
(23)

Die Lösung der inhomogenen linearen Differentialgleichung zweiter Ordnung ist die Summe aus der allgemeinen Lösung der zugehörigen homogenen Gleichung und einer speziellen Lösung des inhomogenen Falls ([8]).

$$\varphi(t) = a_0(\omega) \cdot \sin(\omega t + \varepsilon) + a_1 \cdot e^{-\frac{b-b^*}{2\theta}t} \cos(\varpi t) + a_2 \cdot e^{-\frac{b-b^*}{2\theta}t} \sin(\varpi t)$$
(24)

Für große Werte von t dominiert der erste Summand, das Pendel schwingt mit der Anregungsfrequenz  $\omega$  und der Amplitude  $a_0(\omega)$ .

$$a_{0}(\omega) = \frac{T_{0}^{*}}{\sqrt{(T_{r,max} - \theta\omega^{2})^{2} + (b - b^{*})^{2}\omega^{2}}} \qquad \lim_{\omega \to 0} a_{0} = \frac{T_{0}^{*}}{T_{r,max}} \qquad \lim_{\omega \to +\infty} a_{0} \to 0 \quad (25)$$

Bei der Resonanzfrequenz  $\omega_{max}$  nimmt  $a_0(\omega)$  für  $\frac{T_{r,max}}{\theta} > \frac{1}{2} \left(\frac{b-b^*}{\theta}\right)^2$  ihr Maximum an.

$$\omega_{max} = \sqrt{\frac{T_{r,max}}{\theta} - \frac{1}{2} \left(\frac{b - b^*}{\theta}\right)^2}$$
(26)

<sup>&</sup>lt;sup>6</sup>Es wurde der sichtbare Jupiterradius zugrunde gelegt.



Abbildung 22: Die Schwingungsdauer hängt vom Wert der Fallbeschleunigung ab.



Abbildung 23: Die Auslenkung des Pendels hinkt hinter der Anregung hinterher. Die Phasenverschiebung kann aus der Rechengröße  $t_0$  und der Meßgröße  $t_1$  bestimmt werden.

In der Nähe der Resonanzfrequenz wird die Amplitude meist so groß, daß die Voraussetzung der Kleinwinkelnäherung nicht mehr erfüllt ist. In diesem Bereich zeigen die Meßpunkte einen anderen Verlauf als Gleichung 25 und 26, vergleiche Abbildung 24 b.

Für die Phasenverschiebung  $\varepsilon$  der angeregten Schwingung im Vergleich zum anregenden Moment gilt  $-\pi \leq \varepsilon \leq 0$ . Die Formel zur Messung der Phasenverschiebung ist in Abbildung 23 illustriert.

$$\varepsilon = \arctan \frac{(b - b^*) \,\omega}{\theta \omega^2 - T_{r,max}} \tag{27}$$

Je größer der Stokessche Reibungsterm  $b-b^*$  ist, desto flacher wird die Phasenverschiebungs- und Resonanzkurve, vergleiche Abbildung 24*a* und *c*.

Für kleine Zeiten t sind der zweite und dritte Summand aus Gleichung 24 nicht mehr vernachlässigbar. Mit den Anfangsbedingungen  $\varphi(t=0) = 0$  und  $\dot{\varphi}(t=0) = 0$ ergibt sich für  $a_1$  bzw.  $a_2$  Gleichung 28.

$$a_1 = -a_0 \cdot \sin(\varepsilon)$$
  $a_2 = \frac{a_0}{\varpi} \cdot \left(\frac{b_* - b}{2\theta}\sin(\varepsilon) - \omega \cdot \cos(\varepsilon)\right)$  (28)

 $\varpi$  wurde bereits in Gleichung 18 definiert. Liegen die Werte für  $\varpi$  und  $\omega$  nahe beieinander, beobachtet man beim Einschwingvorgang Schwebungseffekte, vergleiche Abbildung 25.



Abbildung 24: Resonanz- und Phasenverschiebungskurven einer erzwungenen Schwingung. Der Verlauf der Kurven hängt von der Größe des Stokesschen Reibungsterms ab. In der Nähe der Resonanzfrequenz ist die Voraussetzung der Kleinwinkelnäherung nicht mehr erfüllt (b).



**Abbildung 25:** a) Nach etwa 15*s* ist der Einschwingvorgang abgeschlossen, das Pendel schwingt mit der anregenden Frequenz und der Amplitude  $a_0 (8, 7Hz) = 0, 11\pi$ . b) zeigt den Verlauf der Einhüllenden. Die Computersimulation der Gleichung 24 in c) weist dieselbe Charakteristik wie die Kurve in *a* auf, die Kleinwinkelnäherung fällt kaum ins Gewicht.

#### 4.5 Von der Periodizität ins Chaos

Die allgemeine Bewegungsgleichung 15 kann für  $c^* = c$  und  $T^*_{r,max} = 0$  in ein System aus drei Differentialgleichungen erster Ordung für die drei Variablen  $\varphi$ ,  $\upsilon := \dot{\varphi}$  und  $\phi := \omega t$  umgewandelt werden ([9]).

$$\dot{\upsilon} = -\frac{b-b^*}{\theta} \cdot \upsilon - \frac{T_{r,max}}{\theta} \cdot \sin(\varphi) + \frac{T_0^*}{\theta} \cdot \sin(\phi)$$
  
$$\dot{\varphi} = \upsilon$$
  
$$\dot{\phi} = \omega$$
  
(29)

Die Lösung dieses Systems ist eine durch die Zeit t parametrisierte Kurve im dreidimensionalen Phasenraum, der von  $\phi$ ,  $\varphi$  und v aufgespannt wird.

Die Dimension des Lösungsraumes N bestimmt die Anzahl der möglichen Lösungstypen ([10]):

- N = 1: Eine Bewegung im eindimensionalen Phasenraum ist immer monoton, die Lösung läuft gegen einen stationären Punkt (auch " $\infty$ " ist in diesem Sinne ein stationärer Punkt). Ein Beispiel hierfür ist der Verlauf der Einhüllenden bei einer gedämpften Schwingung, vergleiche Abbildung 17*b*.
- N = 2: Ein zweidimensionaler Phasenraum ist bei der freien gedämpften Schwingung realisiert, es gilt  $\phi = 0$ . Die Lösung kann wiederum gegen einem stationären Punkt im Phasenraum streben oder periodisch sein. Im periodischen Fall ist die Phasenraumbahn unter Annahme der Kleinwinkelnäherung eine Ellipse, vergleiche Abbildung 21 c. Das Einmünden in einen stationären Zustand kann sowohl monoton als auch oszillatorisch erfolgen, vergleiche Abbildung 19.
- N = 3: Im dreidimensionalen Phasenraum gibt es neben stationär und periodisch als dritten Lösungstyp auch chaotisch. Abbildung 26 zeigt den Verlauf einer dreidimensionalen periodischen Schwingung. Die charakteristischen Merkmale chaotischer Lösungen sind:
  - 1. Für keine Zeit t wird das Verhalten stationär oder periodisch.
  - 2. Die drei Einzellösungen  $\varphi(t)$ ,  $\upsilon(t)$  und  $\phi(t)$  bleiben innerhalb gewisser endlicher Schranken.
  - 3. Der zeitliche Verlauf der Einzellösungen hängt empfindlich von den Anfangsbedingungen ab. Ein beliebig kleiner, nicht meßbarer Unterschied bei den



Abbildung 26: Dreidimensionale Phasenbahn einer angeregten Schwingung nach dem Einschwingvorgang. Ist  $\phi_2 - \phi_1 = 2\pi n \ (n \in \mathbf{N})$ , sind die zugehörigen Werte für  $\varphi$  und v identisch.

Anfangsbedingungen kann nach kurzer Zeit zu einem völlig verschiedenen Verlauf der Phasenbahn führen.

Bei genauer Kenntnis der Anfangsbedingungen ist das Gleichungssystem 29 auch im chaotischen Fall eindeutig numerisch lösbar. In der Praxis können die Anfangsbedingungen jedoch nicht mit beliebiger Genauigkeit gemessen werden, so daß faktisch keine Voraussagen über einen längeren Zeitraum möglich sind. Man spricht von deterministischem Chaos. In Abbildung 27 wurde der Pendelverlauf nach Gleichung 29 numerisch mit dem Computer gelöst und zwei Meßergebnissen gegenübergestellt. Während in Abbildung 25 die Ungenauigkeit der Parameter vernachlässigbar war, führt diese hier zu einem völlig anderem Kurvenverlauf. Selbst bei zwei unmittelbar hintereinander ausgeführten Messungen reicht die Ungenauigkeit der Motornullstelle aus, um den Verlauf der Kurve ab etwa 3 Sekunden unvorhersagbar zu machen.

Eine aufschlußreiche zweidimensionale Darstellung mehrdimensionaler Phasenraumbahnen sind diejenigen Poincaréplots<sup>7</sup>, bei denen die Phasenbahn fortwährend von Ebenen der Form  $\phi = 2\pi n + \xi$  mit n = 0, 1, 2, ... geschnitten wird und die Schnittpunkte dann auf die ( $\varphi$ , v)-Ebene projeziert werden (vergleiche [11]). Im stationären und periodischen Fall erhält man eine endliche, im chaotischen Fall eine unendliche Anzahl von Punkten, vergleiche Abbildung 28 und 29.

<sup>&</sup>lt;sup>7</sup>Henri Poincaré (1854-1912)



Abbildung 27: Chaotisches Verhalten als Simulation und als Messung. Die Ungenauigkeit der Parameter und der Motornullstelle führt zur Unvorhersagbarkeit der Pendelbewegung nach etwa 3 Sekunden.



Abbildung 28: Periodisches (1) und chaotisches Verhalten (2) einer erzwungenen Pendelschwingung: a) Weg-Zeit-Diagramm b) zweidimensionaler Phasenraum c) als Poincaréplot. Bei periodischer Bewegung besteht der Poincaréplot unabhängig von der Dauer der Messung nur aus einem Punkt, im chaotischen Fall wird die Anzahl der Meßpunkte beliebig groß.



**Abbildung 29:** Poincaréplots mit verschiedener Meßphase: a)  $\xi = \frac{1}{4}\pi$ , b)  $\xi = \frac{3}{4}\pi$ , c)  $\xi = \frac{5}{4}\pi$ , d)  $\xi = \frac{7}{4}\pi$ . Bei Variation der Meßphase krümmt sich die aufgenommene Struktur. Die Muster haben fraktalen Charakter, vergleiche [12] und [13]. e) und f) zeigen eine Computersimulation für  $\xi = \frac{1}{4}\pi$  bzw.  $\xi = \frac{7}{4}\pi$  bei doppelter Anregungsamplitude. Eine Messung ist bei diesen Parametern aufgrund der begrenzten Schrittmotorgeschwindigkeit nicht mehr möglich.

## 5 Schlußbetrachtung

Angefangen hat alles mit dem Wunsch von Herrn Prof. Nachtmann, das chaotische Verhalten eines Oszillators im Vorlesungssaal zu demonstrieren. Die Idee, die erforderlichen Drehmomente mit einem computergesteuerten Schrittmotor über eine Drehfeder auf das Pendel zu übertragen, hatte Herr Dr. J. Schemmel. Unter Betreuung von Herrn Prof. Meier wurde das Pendel dann - je im Rahmen einer Staatsexamensarbeit - von Herrn M.Öhmig gebaut und vom Autor optimiert, geeicht und programmiert. Die Grundlagen der Hardwareansteuerung, wie sie in 2.2 beschrieben wurden, basieren dabei auf einem Testprogramm aus [1], mit Herrn M.Öhmig die prinzipielle Funktionstüchtigkeit der Apparatur nachweisen konnte.

Es wurde erreicht, daß sämtliche Parameter der Schwingungsgleichung über Schieberegler von einer grafischen Benutzeroberfläche aus variierbar sind. Der Einfluß der Regelung auf die Pendelbewegung konnte gemessen werden, so daß mit dieser Apparatur durchaus Präzessionsmessungen möglich sind, wie die Versuche zur Schwingungsdauer zeigen. Bei der Bestimmung der Resonanzfrequenz liegen die gemessenen Werte erwartungsgemäß niedriger als die berechneten, da die Kleinwinkelnäherung bei großen Amplituden falsch wird.

Es lassen sich mit diesem Oszillator nicht nur alle Versuche zur Schwingungsgleichung durchführen, für die in Praktika das Pohlsche Rad benutzt wird, es ist auch möglich, das chaotische Verhalten zu demonstrieren, das häufig nach einem Pendelüberschlag zu beobachten ist. Insofern ist dieser Oszillator eine echte Verbesserung gegenüber dem Pohlschen Rad. Ein Pendelüberschlag wird bei der gegebenen Apparatur nur bei einem leichten Pendelkörper und in der Nähe der Resonanzfrequenz erreicht, da durch eine große Anregungsamplitude leicht die Maximalgeschwindigkeit des Motors überschritten wird. Insofern ähneln sich die aufgenommenen Poincaréplots bei gleicher Meßphase. Mit den gegebenen Parametern braucht der Vergleich mit einer Simulation aber nicht gescheut zu werden<sup>8</sup>.

In einem nächsten Schritt wäre eine grundlegende Überarbeitung der Hardware denkbar. Durch eine präzisere Drehfeder, ein höheres Auflösungsvermögen, ein höheres Haltemoment und eine höhere Maximalgeschwindigkeit des Schrittmotors könnte die Präzession des Pendels und das Anwendungsspektrum weiter vergrößert werden. Ein weiteres Ziel könnte die Darstellung der Grafiken in Echtzeit, also direkt während der

<sup>&</sup>lt;sup>8</sup>Im Internet kann unter "http://monet.physik.unibas.ch/~elmer/pendulum/index.html" u. a. eine Simulation des chaotischen Verhaltens eines mathematischen Pendels heruntergeladen werden.

Pendelbewegung, sein. Mit der vorhandenen Software ist dies zwar problemlos realisierbar, die Plotfunktionen sind aber zu langsam, um in der Regelschleife aufgerufen zu werden, deren Durchlaufzeit zwischen 10 ms und 0,77 ms beträgt. Eine Lösung wäre der Bau eines Mikrocontrollers, der über eine serielle Schnittstelle mit einem Computer verbunden wird. Während der Mikrocontroller die komplette Regelung übernimmt, werden die Meßdaten zum Computer geschickt und von der Regelung zeitlich unabhängig dargestellt.

A ANHANG

## A Anhang

## A.1 Sourcecode

Die Programme zur Bedienung des Pendels sind in C++ unter Verwendung der Bibliotheken Qt für die grafische Benutzeroberfläche und Qwt für die Plotfenster geschrieben. Aus Platzgründen ist nur die Regelschleife dokumentiert.

In Abschnitt A.1.2 findet sich der Code zur Simulation des chaotischen Pendelverhaltens mit dem Computer-Algebra-System *Mathematica*.

#### A.1.1 Regelschleife

Der Dialoge zwischen Benutzeroberfläche und den Funktionen zur Regelung, grafischen Darstellung und Datenspeicherung erfolgt über die Variable *modus*. Bei Programmstart und nach Beendigung sämtlicher Funktionen wird dieser Variablen der Wert Null zugewiesen.

- **modus=0:** Die Regelschleife läuft nicht, sie kann mit der **Start**-Taste aufgerufen werden.
- modus=1: Die Regelschleife läuft, ein doppelter Funktionsaufruf ist nicht möglich.
- modus=2: Die Regelschleife wird abgebrochen, um das Pendel in die Nullposition zu fahren. Die zugehörige Funktion ist eine modifizierte Regelschleife, in der auf die bei Programmstart gespeicherte Nullage geregelt wird.
- modus=3: Die Regelschleife wird abgebrochen, um die aufgenommenen Meßdaten grafisch darzustellen und abzuspeichern. Dies erfolgt entweder, wenn die angegebene Anzahl der Meßwerte erreicht oder die Stop bzw. Stop/Pendel auf Null fahren -Taste gedrückt wurde.
- **modus=4:** Die **Schließen** Taste wurde gedrückt. Der allokierte Speicherbereich wird freigegeben und das Programm beendet.
- modus=5: Die Regelschleife wird ohne weiteren Funktionsaufruf abgebrochen.
- modus=7: Das Pendel wird in die Nullposition gefahren, in dieser Zeit werden alle anderen Funktionsaufrufe ignoriert.

## A.1 Sourcecode

#### Regelschleife von pendeldemo

if(modus==0)					
(modus=1; //Schützt vor doppeltem Funktionsaufruf					
//Variablen					
int apos;	//Drehgeberwert				
double ppos; //Pendelposition von 0 bis 8192					
double pposn;	ouble pposn; //Pendelposition in pi				
<b>double</b> $pposv=0;$	ouble pposv=0; //Vergleichswert für pposn				
double $w=0;$	ouble $w=0$ ; //Winkelgeschwindigkeit in $\pi/s$				
double y=0;	//Größe der Regelabweichung				
<b>double</b> $y1=0;$	//Größe der Regelabweichung				
<b>double</b> $x=0;$	//Winke  zwischen Motor und	Pendel			
double $t=0;$	//Echtzeitvariable [s]				
$\mathbf{double}\; \mathrm{tv}{=}0;$	//Vergleichswert für t				
double k;	//1/k ist die Wartezeit in s				
double $kv = 100;$	//Vergleichswert für k				
double step;	//Schritte des Taktgebers				
double correct;	//Variable für die Korrektur de	es Richtmoments			
int n=0;	//Zählt die Schleifendurchläufe				
int update=0;	//Wird für Benutzeroberflächer	aktualisierung benötigt			
$\mathbf{double} \; \mathrm{mspeed}{=}0;$	//Regelgeschwindigkeit in Schr	itte/Sekunde			
<pre>int vorz=0;</pre>	//Vorzeichen bei Dämpfungstermen				
int $rst=0;$	//Anzahl der aufgenommenen N	Леßwerte			
int werte;	//Anzahl der aufzunehmenden I	Neßwerte bei Start der Regelschleife			
int wert;					
//Inbetriebnahme des	Taktgebers				
LARGE_INTEGER h	ocfreq;	//hier steht die Frequenz des Taktgebers			
LARGE_INTEGER la	sttime;	//Zeitvariablen			
LARGE_INTEGER ne	ewtime;				
$LARGE_INTEGER$ st	artzeit;				
LARGE_INTEGER ne	euzeit;				
${\it Query Performance Free}$	quency(&hpcfreq);	//Abfrage der Frequenz des Taktgebers			
//Anlegen einer dynam	ischen Datenstruktur zur Aufnah	me der Meßwerte			
delete[]parray;					
$werte = spin1 \rightarrow value();$					
parray = new messung	[werte];				
asm cli;	//Verbot der betriebssysteminte	rnen Interrupts			
//Regelschleite					
while(1)					
{ if(n==0){Query PerformanceCounter(&startzeit); //Abfrage der Startzeit					
Query PerformanceCounter (& new time);					
$Query PerformanceCounter(\& last time); \}$					
// A: Berechnen der Echtzeit					
$\mathbf{if}(\mathbf{n}!=0)$					
{					
QueryPerformanceCounter(&neuzeit);					
t = (neuzeit.QuadPart-startzeit.QuadPart)/double(hpcfreq.QuadPart);					
mspeed = 1/(t-tv):	••••••••••••••••••••••••••••••••••••••	· · //			
}					
ſ					

//B: Auslesen des Drehgebers, berechnen der Pendelposition in  $\pi$ apos=position\_DG(mact);//Auslesen des Drehgebers; mact speichert die aktuellen Motorkonditionenppos=first-apos;pposn=ppos/4096;if(pposn>=1)pposn==2;//Pendelwerte bei Überschlagif(pposn<-1)pposn+=2;</td>

#### $//{\mathbf{C}}:$ Bewegen des Motors um einen Schritt

if(y>=0.005)DIG\_Out\_Port(DivNum,PA,(mact|DG\_TAKTM)& M\_CLOCKM);

#### $//\mathbf{D}$ : Berechnen der Geschwindigkeit $\mathbf{if}(n!=0)$

{w=(pposn-pposv)/(t-tv); if(w>10)w=10; //Unterdrückung von Ausreißern if(w<-10)w=-10; if(w>0)vorz=1; if(w<0)vorz=-1; if(w==0)vorz=0;}

#### $// {\bf E}:$ Aufnahme der Daten in den Arbeitsspeicher

if(box1→isChecked()==TRUE)
{parray[rst].Pendelposition=pposn;
parray[rst].Pendelgeschwindigkeit=w;
parray[rst].Regelabweichung=y1;
parray[rst].Winkel=x;
parray[rst].Echtzeit=t;
parray[rst].Wartezeit=(t-tv);
parray[rst].Motorgeschwindigkeit=mspeed;
parray[rst].Motorposition=mpos;
rst++;}

#### $//\mathbf{F}$ : Aktualisieren der Benutzeroberfläche

asm sti; //Betriebssysteminterrupts zugelassen if(tv<=update/10.0 && t>update/10.0){qApp→processEvents(1); //jede Zehntelsekunde wird die update++;} //Benutzeroberfläche akualisiert //Abbrechen der Regelschleife in verschiedenen Modi if(modus==2)break; **if**(modus==4)break; **if**(spin1→value()>=werte)wert=werte; else wert=spin1 $\rightarrow$ value();  $if(box1 \rightarrow isChecked() == TRUE \&\& rst >= wert) \{modus=3;$ break;} if(modus==5)break; //keine Betriebssysteminterrupts asm cli;

#### $//{\bf G} {:}\ {\rm Berechnen}\ {\rm der}\ {\rm neuen}\ {\rm Regelabweichung}\ y$

$$\label{eq:correct} \begin{split} & \operatorname{correct} = -\operatorname{lcd5} \rightarrow \operatorname{value}()/180.0^* \sin(\operatorname{lcd6} \rightarrow \operatorname{value}()^*t) - \operatorname{vorz}^* \operatorname{lcd2} \rightarrow \operatorname{value}()/3.94^* 0.005 - \operatorname{lcd3} \rightarrow \operatorname{value}()/1.25^* 0.005^* \mathrm{w} \\ & -\operatorname{lcd4} \rightarrow \operatorname{value}()/3.94^* 0.005^* \sin(\operatorname{pposn}^* \mathrm{M\_PI}); \\ & \operatorname{if}(\operatorname{correct} > 0) \operatorname{correct} = (0.250812/0.253858)^* \operatorname{correct}; \\ & \operatorname{else} \ \operatorname{correct} = (0.250812/0.247766)^* \operatorname{correct}; \\ & \operatorname{y=correct} - (\operatorname{pposn} - \operatorname{mpos}); \end{split}$$

#### A.1 Sourcecode

 $\label{eq:linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_linear_line$ 

$$\label{eq:mpos} \begin{split} &mpos = mpos + 0.005; \} \\ & \mathbf{if}(mpos > = 1)mpos = = 2; \\ & \mathbf{if}(mpos < -1)mpos + = 2; \end{split} \qquad \qquad //\mathsf{Motorwerte bei } \ddot{\mathsf{U}}\mathsf{berschlag} \end{split}$$

#### //I: Berechnen der Restwartezeit

$$\begin{split} &y{=}fabs(y); \\ & if(y{<}{=}0.005)k{=}lcd7{\rightarrow}value(); \\ & else\ k{=}1000*lcd1{\rightarrow}value()*(y{-}0.005){+}lcd7{\rightarrow}value(); \end{split}$$

#### //Beschleunigungsbegrenzung

if(box2→isChecked()==TRUE)
{if(k>kv && k\*kv-kv\*kv>lcd8→value())
k=lcd8→value()/kv+kv;
if(k<kv && kv\*kv-k\*kv>lcd8→value())
k=kv-lcd8→value()/kv;}
if(k>1300)k=1300;
kv=k;
tv=t;
pposv=pposn;
n++;
step=hpcfreq.QuadPart/k;
//Warteschleife
while(newtime.QuadPart-lasttime.QuadPart<step)
{QueryPerformanceCounter(&newtime);}
lasttime.QuadPart=newtime.QuadPart;}</pre>

//Beschleunigungsbegrenung bei ansteigenden
//Geschwindigkeiten
//Beschleunigungsbegrenung bei fallenden
//Geschwindigkeiten

//Maximale Motorgeschwindigkeit

#### //Funktionsaufruf nach Verlassen der Regelschleife

```
if(modus==5){if(box1→isChecked()==TRUE)modus=3;
else modus=0;}
if(modus==3)
{dat=rst;
plotten(dat); //Aufruf der Plotfunktion
datenspeicherung();} //Speicherung der Daten
if(modus==2)relax(); //Funktion zur Drehung des Pendels in die Nullage
if(modus==4){delete[]parray;
DIG_Out_Port(DivNum,PA,0);
qApp→exit(0);} //Funktion zur Beendigung der Applikation
}
```

Die Punkte **A** bis **I** entsprechen der in Abschnitt 2.3.2 gemachten Einteilung. Die Variable *correct* bezeichnet den Soll-Wert  $\psi$ , vergleiche Gleichung 14, der Ist-Wert  $\tilde{\psi}$  ist die Differenz aus der Pendelposition *pposn* und der Motorposition *mpos. box1* ist der Zeiger auf die Kontrollbox "Aufnahme von Meßwerten", *box2* auf "Maximale Motorbeschleunigung". Der Zugriff auf die Werte der LCD-Anzeigen erfolgt über  $lcd\Box$ , wobei teilweise noch Umrechnungen vorgenommen werden müssen.

Bei den Programmen "resonanz" und "chaos" unterscheidet sich die Regelschleife in erster Linie in den Punkten **D** und **E**. Bei "resonanz" wird in der letzten Anregungsamplitude die Amplitude des Pendels und die Zeit des letzten Nulldurchgangs mit negativer Flanke bestimmt, vergleiche Abbildung 23.  $lcd4 \rightarrow value()$  ist die Kreisfrequenz  $\omega$ ,  $lcd6 \rightarrow value()$  der Wert der Erhöhung der Kreisfrequenz und  $spin2 \rightarrow value()$ die Konstante C aus Gleichung 11.

#### resonanz

```
//D: Berechnen der Geschwindigkeit und der gewünschten Daten
//Bestimmung der Amplitude in der letzten Periode
if(t \ge (spin 2 \rightarrow value()*ceil(lcd 4 \rightarrow value())-1)*2*M_PI/lcd 4 \rightarrow value())
{if((pposv>0 && pposn>0) && pposv<pposn)amplitude=pposn;}
//Bestimmung der Zeit des letzten Nulldurchgangs mit negativer Flanke
if(t > (spin2 \rightarrow value())*ceil(lcd4 \rightarrow value())-1)*2*M_PI/lcd4 \rightarrow value())
{if((pposv>=0 && pposn>=0) && pposv>pposn)tp2=t;}
//E: Aufnahme der Daten in den Arbeitsspeicher
if(t > spin2 \rightarrow value()*ceil(lcd4 \rightarrow value())*2*M_PI/lcd4 \rightarrow value())
{
parray[a].Amplitude=amplitude;
parray[a].Kreisfrequenz=lcd4\rightarrowvalue();
parray[a]. Phase = (spin2 \rightarrow value()*ceil(lcd4 \rightarrow value())*2*M_PI/lcd4 \rightarrow value()-tp2)*lcd4 \rightarrow value()/M_PI-1;
a++;
QueryPerformanceCounter(&startzeit);
t = 0:
tv=0;
update=0;
slid=slider6 \rightarrow value()+slider4 \rightarrow value();
                                                                              //Berechnung des neuen Schiebereglerwertes
if(slid>141)slid=141;
emit omegaplus(slid);
                                      //Zuweisung der neuen Kreisfrequenz an den Schieberegler
}
```

Das Programm "chaos" mißt nach dem Einschwingvorgang die Pendelposition und die Pendelgeschwindigkeit zu den Zeiten  $t = \frac{2\pi n + \xi}{\omega}$   $(n \in \mathbf{N})$ .  $spin2 \rightarrow value()$  gibt die Anzahl der Perioden des Einschwingvorgangs an,  $lcd4 \rightarrow value()$  die Anregungsfrequenz  $\omega$  und  $lcd6 \rightarrow value()$  die Meßphase  $\frac{\xi}{2\pi}$ .

chaos

```
//E: Aufnahme der Meßwerte
if( (tv<=(spin2→value()+lcd6→value()+rst)*2*M_PI/lcd4→value())
&& (t>=(spin2→value()+lcd6→value()+rst)*2*M_PI/lcd4→value()) )
{parray[rst].Pendelposition=pposn;
parray[rst].Pendelgeschwindigkeit=w;
rst++;}
```

#### A.1.2 Simulation des chaotischen Pendelverhaltens

Zur Bestimmung des Weg-Zeit-Diagramms wird für Gleichung 15 mit  $c^* = c$  und  $T^*_{r,max} = 0$  eine Wertetabelle von  $t_{min}$  bis  $t_{max}$  in Schritten von  $t_{step}$  erstellt. Der Wertebereich der Pendelposition ist unabhängig von der Zahl der Überschläge  $-\pi \leq \varphi < \pi$ . Die Funktion f bewirkt eine Projektion der berechneten Werte der Pendelposition  $\varphi$  in dieses Intervall. Da in Gleichung 15 mit den gemachten Einschränkungen nur der Sinuswert und die Ableitung von  $\varphi$  eingehen, ist es gerechtfertigt, diese Angleichung an das reale Pendelverhalten erst nach der numerischen Lösung vorzunehmen.

Weg-Zeit-Diagramm des chaotischen Pendelverhaltens

 $\begin{array}{l} //1: \mbox{ Definition der Projektionsfunktionen f} \\ f[x\_] := x \ /; \ (-\pi \leq x < \pi); \\ f[x\_] := x - \left( \mbox{Round } \left[ \frac{x}{2\pi} \right] 2\pi \right) \ /; \ (x \geq \pi \ \parallel \ x < -\pi); \end{array}$ 

```
\begin{array}{ll} //2: \mbox{Eingabe der Parameter} \\ \theta = \ldots; \\ T^*_{r,max} = \ldots; \\ T^0_0 = 0.251 \frac{\pi}{180} \ldots; \\ \omega = \ldots; \\ b = \ldots; \end{array} // \mbox{Anregungsamplitude im Gradmaß} \\ \end{array}
```

$$\begin{split} t_{min} &= \dots; \\ t_{max} &= \dots; \\ t_{step} &= \dots; \\ step_{max} &= \dots; \end{split} \ // \mathsf{Defaultwert ist 1000} \end{split}$$

#### //3: Erstellen und Speichern der Wertetabelle

$$\begin{split} &\text{liste1}=\text{Table}[\text{Evaluate}[\varphi[t] \ /. \ \text{NDSolve}[\{\theta\varphi''[t]+T^*_{r,max}\text{Sin}[\varphi[t]]+b\varphi'[t]-T^*_0\text{Sin}[\omega t]==0, \ \varphi[0]==\varphi'[0]==0\}, \ \varphi, \ \{t, t_{min}, t_{max}\}, \ \text{MaxSteps}\rightarrow step_{max}]], \ \{t, t_{min}, t_{max}, t_{step}\}]; \\ &\text{liste2}=\text{Table}[\{t_{step}(i-1), \ \text{TableForm}[\text{liste1}[[i]], \ \text{TableDepth}\rightarrow 2]\}, \ \{i, 1, (t_{max}-t_{min})/t_{step}+1\}]; \\ &\text{Export}["\ldots] \\ &\text{Liste.dat", liste2];} \end{split}$$

//4: Projektion der Pendelposition arphi und Plotten

 $liste3=Import["... \liste.dat"]; liste4=MapAt[f, liste3, Table[{i, 2}, {i, 1, (t_{max} - t_{min})/t_{step} + 1}]]; ListPlot[liste4]$ 

Die Simulation der Poincaréplots erfolgt über das Gleichungssystems 29. Statt der Grenzen und der Auflösung einer Wertetabelle werden die Anzahl der aufzunehmenden Punkte n und die Meßphase  $\frac{\xi}{2\pi}$  angegeben.

#### Simulation eines Poincaréplots

 $\begin{array}{l} //1\text{: Definition der Projektionsfunktionen f} \\ f[x_{-}] := x \ /; \ (-\pi \leq x < \pi); \\ f[x_{-}] := x - \left( \text{Round} \left[ \frac{x}{2\pi} \right] 2\pi \right) \ /; \ (x \geq \pi \ \parallel \ x < -\pi); \end{array}$ 

 $\begin{array}{ll} //2: \mbox{Eingabe der Parameter} \\ \theta = \dots; \\ T^*_{r,max} = \dots; \\ T^*_0 = 0.251 \frac{\pi}{180} \dots; \\ \omega = \dots; \\ b = \dots; \\ n = \dots; \\ \xi = 2\pi \dots; \\ step_{max} = \dots; \\ \end{array} // \mbox{Defaultwert ist 1000} \end{array}$ 

 $\begin{aligned} //3: & \text{Berechnen der Pendelposition } \varphi \text{ und der Pendelgeschwindigkeit } v \text{ bei } t = \frac{2\pi n + \xi}{\omega} \ (n \in \mathbb{N}) \\ & \text{listel=Table[Evaluate[}\{\varphi[t], v[t]\} \ /. \ \text{NDSolve[}\{\varphi'[t] == \omega, \varphi'[t] == v[t], v'[t] == -\frac{b}{\theta}v[t] - \frac{T_{r, max}^*}{\theta}\text{Sin}[\varphi[t]] + \frac{T_0^*}{\theta}\text{Sin}[\phi[t]], \\ & \phi[0] == v[0] == \varphi[0] == 0], \ \{\phi, \varphi, v\}, \ \{t, \frac{\xi}{\omega}, \ (n-1)\frac{2\pi + \xi}{\omega}\}, \ \text{MaxSteps} \rightarrow step_{max}]], \ \{t, \frac{\xi}{\omega}, \ (n-1)\frac{2\pi + \xi}{\omega}, \frac{2\pi}{\omega}\}]; \\ & \text{liste2=Table[TableForm[liste1[[i]], TableDepth \rightarrow 2]}, \ \{i, 1, n\}]; \\ & \text{Export["... \\liste.dat", liste2];} \end{aligned}$ 

//4: Projektion der Pendelposition  $\varphi$  und Plotten liste3=Import["...\liste.dat"]; liste4=MapAt[f, liste3, Table[{i, 1}, {i, 1, n}]]; ListPlot[liste4]

## A.2 Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die zum Gelingen dieser Arbeit beigetragen haben, insbesondere bei

- Herrn Prof. Dr. K. Meier für die Bereitstellung des interessanten Themas und die motivierende Betreuung.
- Herrn Dr. J. Schemmel für die geduldige Unterstützung in der Einarbeitungsphase, die wertvollen Anregungen und Tips bei der Programmierung und für die Durchsicht des Skripts.
- Herrn T. Maucher für ein offenes Ohr bei Problemem aller Art sowie hilfreichen Verbesserungsvorschlägen und Korrekturen beim Zusammenschreiben.
- Herrn A. Breidenassel und Herrn C. Schuhmacher.
- den Mitarbeitern des Kirchhoff-Instituts für die organisatorische Unterstützung und die Bereitstellung der notwendigen Infrastruktur.
- den Mitarbeitern der Elektronikwerkstatt, insbesondere Herrn Uhlsen, für die Überarbeitung der elektronischen Schnittstelle.
- den Mitarbeitern der mechanischen Werkstatt für die Fertigung bzw. Überarbeitung der mechanischen Komponenten.

Dieses Dokument wurde mit dem Textverarbeitungssystem LATEX erstellt.

#### A.3 Literaturverweise

## Literatur

- [1] M.Öhmig, "Bau und Test eines chaotischen Pendels mit Wirbelstrombremse", Wissenschaftliche Arbeit Mai 2000
- [2] Redaktion Naturwissenschaft und Technik des Bibliographischen Instituts (Hrg.), "Wie funktioniert das? - Die Technik im Leben von heute", Meyers Lexikonverlag, Mannheim 1978
- [3] Wolfgang Nolting, "Grundkurs Theoretische Physik 1 Klassische Mechanik", Viewegverlag, Wiesbaden 1997
- [4] Wolfgang Nolting, "Grundkurs Theoretische Physik 2 Analytische Mechanik", Viewegverlag, Wiesbaden 1998
- [5] Christian Gerthsen/Helmut Vogel, "Physik", Springer-Verlag, Berlin 1993
- [6] Paul A. Tipler, "Physik", Spektrum Akademischer Verlag, Heidelberg 1994
- [7] http://monet.physik.unibas.ch/~elmer/pendulum/nonosc.htm
- [8] H. Fischer/H. Kaul, "Mathematik für Physiker 1", Teubner-Verlag, Stuttgart 1988
- [9] G.L. Baker, J.P. Gollub, "Chaotic dynamics an introduction", Cambridge University Press, New York 1996
- [10] S.Großmann, "Mathematischer Einführungskurs für die Physik", Teubnerverlag, Stuttgart 1993
- [11] http://monet.physik.unibas.ch/~elmer/pendulum/bterm.htm#pmap
- $[12] http://monet.physik.unibas.ch/\sim elmer/pendulum/chaos.htm$
- $[13] http://monet.physik.unibas.ch/\sim elmer/pendulum/fractal.htm$
- [14] M.K.Dalheimer, "Programming with Qt", O'REILLY Verlag, Köln 1999
- [15] U.Kaiser, "C/C++ Von den Grundlagen zur professionellen Programmierung", Galileo Computing, Bonn 2000

## A.4 Erklärung

"Ich erkläre, daß ich die Arbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und daß alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angabe der Quellen als Entlehnung kenntlich gemacht worden sind."