

Studienarbeit

Sensomotorische Erzeugung eines Gangmusters für humanoide Roboter

Autor: Benjamin Werner
Betreuer: Dr. Manfred Hild

Lehrstuhl für Künstliche Intelligenz
Prof. Dr. Hans-Dieter Burkhard

November 2008



Abstract Die vorliegende Arbeit befasst sich mit der Erstellung von Bewegungsmustern von humanoiden Robotern. Die Implementierung der Bewegungen erfolgt auf den Robotern der A-Serie des NRLs mit Hilfe der Programme MotionEditor und SML. Diese sind im Rahmen der Studienarbeit von Christian Thiele erstellt worden. Insbesondere wird ein künstliches neuronales Netz vorgestellt, welches Vorwärtsgang mit einstellbaren Kurvenradien ermöglicht. Das Gangmuster wird mit Hilfe einer sensomotorischen Schleife erzeugt und mit verschiedenen Neuromodulen verändert um das gewünschte Verhalten zu erreichen.

Inhaltsverzeichnis

1	Einleitung	5
1.1	Verwendete Roboterplattform	5
2	Methoden zur Erzeugung von Bewegungsmustern	8
2.1	Keyframe-Technik	9
2.1.1	Diskussion der Keyframe-Technik	10
2.2	Central-Pattern-Generator	11
2.2.1	Biologische Motivation	11
2.2.2	Neuronenmodell von Kiyotoshi Matsuoka	12
2.2.3	Diskussion des neuronalen Central-Pattern-Generators	13
3	Neuromodule	16
3.1	Neuronenmodell	16
3.1.1	Transferfunktion	17
3.1.2	Neuronales Netz	18
3.1.3	Biasneuron	19
3.1.4	Matrix-Schreibweise eines neuronalen Netzes	19
3.1.5	Vereinfachte Darstellung	19
3.2	Neuromodule mit einem Neuron	20
3.2.1	Schwellwertschalter	20
3.2.2	Neuronale Filterung	20
3.2.3	Hysterese-Neuron	23
3.3	Neuro-Module mit mehreren Neuronen	23
3.3.1	Boolesche Operatoren	24
3.3.2	Gate-Schaltung	25
3.3.3	Linearer Multiplikator	26
3.3.4	Neuronale Oszillatoren	27
3.3.5	Neuronaler $SO(2)$ -Oszillator	30
4	Konstruktion des neuronalen Netzes	32
4.1	Neuronale Reflexschleife zur Erzeugung der Grundschiwingung	32
4.1.1	Konstruktion der neuronalen Reflexschleife	32
4.1.2	Analyse der Grundschiwingung	34
4.1.3	Einfluss der Netzparameter auf die Schwiwingung	35
4.2	Struktur und Beschreibung des neuronalen Netzes	38
4.2.1	Main Driver	38

4.2.2	Interner Oszillator	39
4.2.3	Mischpult	40
4.2.4	Einkopplung der Vorwärtsbewegung	40
4.2.5	Steuerung der Links-Rechts-Rotation	41

1 Einleitung

Humanoide Roboter bilden ein großes Feld der aktuellen Forschung. Es gibt noch immer viele offene Fragen: Welches sind die besten Aktuatoren? Welche Materialien sollen verwendet werden? Wie wird die Energieversorgung sichergestellt? Eins der größten Probleme, das spätestens nach der Konstruktion eines humanoiden Roboters auftritt, ist die Erzeugung stabiler Bewegungen. Im Rahmen dieser Studienarbeit werden zuerst verschiedene Methoden zur Erzeugung von Bewegungsmustern erläutert. Beginnend mit der Darstellung der Keyframe-Technik, die auch im "Neurobotics Research Laboratory" (NRL) zur Anwendung kommt, werden anschließend die neuronalen Central-Pattern-Generatoren beleuchtet, mit deren Hilfe zyklische Bewegungsmuster erzeugt werden können. Dabei wird auch auf ihre adaptive Fähigkeit mit Hilfe von Sensor-Feedback eingegangen. Als zweites werden verschiedene Neuromodule eingeführt. Mit deren Hilfe wird ein neuronales Netz aufgebaut, das das Gehen in Links- und Rechtskurven mit veränderbaren Radien ermöglicht. Dies wird im dritten Kapitel ausführlich behandelt. Hierbei wird zuerst die sensomotorische Schleife behandelt, die das Grundpattern des Ganges erzeugt. Daraufhin folgt eine ausführliche Beschreibung der Struktur und Funktionsweise des gesamten neuronalen Netzes. Abgerundet wird diese Arbeit mit einem prägnanten Fazit und einem weiterführenden Ausblick.

1.1 Verwendete Roboterplattform

Die als A-Serie bezeichnete humanoide Roboterplattform des NRL wurde auf der Grundlage des Roboterbaukastens Bioloid von der Firma Robotis aufgebaut. Der Roboter ist ca. 47 cm hoch, das Gewicht beträgt 2,2 kg und er verfügt über 21 Freiheitsgrade. Die A-Serie besitzt acht so genannte „AccelBoards“. Diese beinhalten einen RISC-Prozessor und einen 2-achsigen Beschleunigungssensor.

- Aktuatoren: Jedes Gelenk des Roboters wird von einem Dynamixel AX-12 bzw. dem Nachfolger AX-12+ angetrieben. Diese Motoren enthalten einen Winkelencoder und werden über ein Hochgeschwindigkeits-Bussystem angesteuert.
 - Masse: 55 g
 - Übersetzungsverhältnis des Getriebes 1/254
 - Maximales Haltedrehmoment 165 Ncm
 - Maximale Geschwindigkeit 0.196s/60

– Auflösung des Winkelencoders: 0.35°

- Sensoren: Zusätzlich zu den Winkelencodern besitzt die A-Serie eine Kamera und acht 2-achsige Beschleunigungssensoren vom Typ ADXL213, die über den Körper verteilt sind. Die Anordnung ist schematisch in der Abbildung 1.2 dargestellt.
- CPUs: Der PDA enthält einen Prozessor des Typs PXA272 von Intel. Dieser verarbeitet die Bilder der Kamera und übernimmt die höheren kognitiven Aufgaben.
- Acht RISC-Microcontroller der Firma Renesas kommen auf den AccelBoards zum Einsatz. Jeweils ein AccelBoard steuert drei Dynamixel an. Die acht AccelBoards kommunizieren über den so genannten "SpinalCord". So liegen zu jedem Zeitschritt jedem AccelBoard alle Daten des Roboterkörpers vor. Die Taktrate des SpinalCord beträgt 100 Hz. Die AccelBoards übernehmen die Körpersteuerung. Mehr Informationen zu der Firmware der AccelBoards finden sich in der Studienarbeit von Christian Thiele [Thi07].



Abbildung 1.1: *Anita Roboter der A-Serie*

Für alle Experimente wird diese Roboterserie verwendet. In der Zeit, als diese Arbeit entstand, machte diese Roboterserie eine Entwicklung durch. Es wurden nach und nach Teile des Bausatzes durch selbst konstruierte Bauelemente, wie zum Beispiel der Oberschenkel, ersetzt. Hierdurch konnte gezeigt werden, dass mit geringer Änderung das neuronale Laufen auch auf der veränderten Version zu einem stabilen Gangmuster führt.

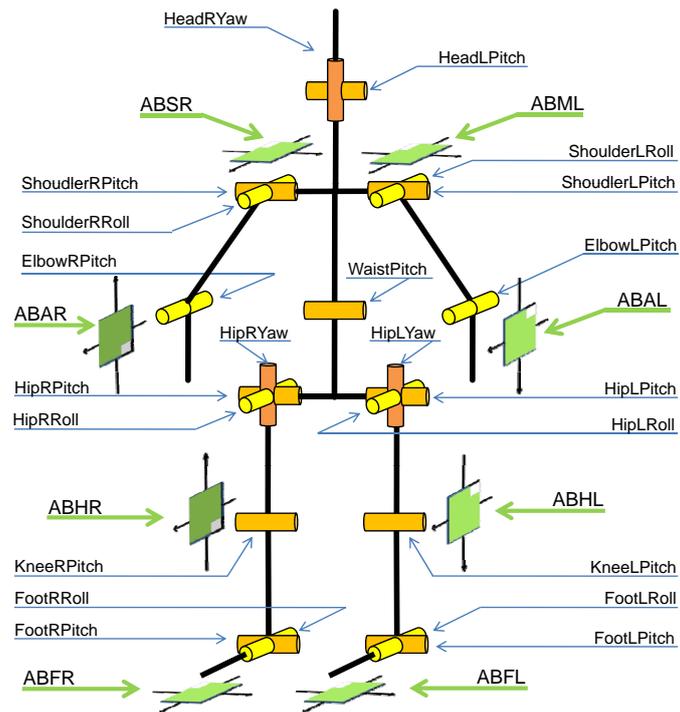


Abbildung 1.2: Schematische Darstellung eines A-Serie Roboters. Die grünen Platten stellen AccelBoards da. Die Beschleunigungsachsen sind als Pfeile eingezeichnet.

2 Methoden zur Erzeugung von Bewegungsmustern

Es gibt viele Methoden, die Bewegungen bei Robotern erzeugen können. Generell können die Methoden in offline und online berechnete Bewegungsmuster eingeteilt werden. Jedoch finden sich auch viele Mischformen, in denen offline erstellte Pattern online angepasst werden.

Offline bedeutet hier, dass die Bewegungstrajektorien zuerst berechnet und dann auf dem Roboter abgespielt werden. Bei der Berechnung der Trajektorien werden verschiedene

physikalische Modelle, z.B. das Inverse Pendel [Kat73] und verschiedene Stabilitätskriterien verwendet. Sehr oft wird hierbei der ZMP herangezogen [Pra00]. ZMP (*ZeroMomentPoint*) definiert den Punkt innerhalb des Support-Polygons, an dem die Ground Reaction Force ansetzt und die vertikalen Momente der auf den Körper wirkenden Kräfte sich ausgleichen. Solange dieser Punkt existiert, das heißt dieser sich innerhalb des Support-Polygons befindet, ist der Roboter stabil [Vuk04]. Bei Online-Verfahren hingegen wird zu jedem Zeitschritt des Systems der nächste gewünschte Zustand berechnet. Auch hier kommen wieder unterschiedliche Verfahren zum Einsatz. Bisher dominiert der ZMP als Kriterium zur Stabilisation des Roboters. So benutzte Asimo von Honda [KH], SDR-4XII von Sony [YK03] und HRP von Aist [KK02] den Target-ZMP-Algorithmus. Jedoch hat dieser einige markante Nachteile. So ist zur Berechnung des ZMPs eine sehr präzise physikalische Modellierung des Roboters nötig. Auch sind hohe Anforderungen an die Aktuatoren und Sensoren gestellt. Die müssen in der Lage sein schnell und sehr präzise eine Position anzufahren [GE04]. Vergleicht man es mit der Fauna, ist es kaum vorstellbar, dass Tiere die aufwendigen ZMP-Berechnungen während der Lokomotion durchführen. Genauso wenig gehört zu den herausragenden Eigenschaften der Muskeln das millimetergenaue Anfahren bestimmter Positionen. Ihre Stärke liegt in der Elastizität und ihren Dämpfungseigenschaften. Fortschritte in der Biologie, die es ermöglichen Nervenzellen intakt zu extrahieren, ihre Aktivität zu messen und ihre Struktur zu analysieren führten zu neuen Erkenntnissen in der neurobiologischen Regelungstechnik. So erfreuen sich neuronale Netze zur Bewegungssteuerung von Robotern immer größerer Beliebtheit. Diese Controller bestehen meist aus neuronalen Oszillatoren, die über Sensor-Feedback an die Umwelt angepasst werden. Zusätzlich werden sie mit so genannten Reflex- oder Sensormotorschleifen erweitert. Das Prinzip ist vereinfacht zu verstehen als ein Oszillator, Central-Pattern-Generator, der eine Schwingung erzeugt, die das Grundpattern einer Bewegung darstellt. Dieses wird

über Sensor-Feedback so angepasst, dass die Bewegung in der Umwelt erfolgreich ausgeführt werden kann. Im Folgenden werden zwei Verfahren genauer betrachtet, die beide keine Modellbildung des Roboters und der Umwelt erfordern. Dies ist zum einen die Keyframe-Technik als Vertreter der Offlinetechniken und zum anderen das Prinzip des Central-Pattern-Generators als Onlinetechnik.

2.1 Keyframe-Technik

Die Idee dieser Bewegungserzeugung ist der Trickfilmtechnik entlehnt. Die Bewegung wird in einzelne Körperposen aufgespalten. Die Bewegung entsteht durch das der Reihe nach Anfahren der einzelnen Posen. Diese Art der Erzeugung von Bewegungspattern erfordert keine physikalisch-mathematische Modellbildung des Roboters. Für diesen Zweck wurde im NRL der MotionEditor entwickelt. Eine detaillierte Beschreibung dessen findet sich in [Thi07]. Eine Bewegung besteht aus so genannten Keyframes. Ein Keyframe ist ein Vektor, der die Winkelpositionen aller Gelenke enthält. Somit entspricht dieser einer Pose des Roboters. Der MotionEditor erlaubt es, die Motoren des Roboters weich zu schalten und dadurch den Körper zu formen. Auf Knopfdruck können die Motoren wieder starr geschaltet werden, dabei werden die Winkelpositionen ausgelesen und in einem Keyframe gespeichert. Die Keyframes werden über Transitionen verbunden. Zwischen einzelnen Winkelwerten der Keyframes wird innerhalb der Transition interpoliert. Dabei besteht die Möglichkeit die Zeitkonstante der Interpolation und das Drehmoment der Motoren einzustellen. Eine Abfolge von Keyframes entspricht nun einer Bewegung. Im Rahmen des Robocup wurde im NRL eine kleine Bibliothek von Bewegungen erstellt. Verschiedene, isolierte Bewegungen können nun zu einem so genannten Motionnet zusammengefasst werden. Dabei gibt es verschiedene Transitionen zwischen den einzelnen Bewegungsabläufen, die mit Hilfe von Selektoren angesprungen werden können. Im folgenden Bild ist ein Motionnet zu sehen, welches dem Roboter ermöglicht, links und rechts zu laufen, sich auf der Stelle links und rechts zu drehen, auf dem Bauch liegend aufzustehen, sich vom Rücken auf den Bauch zu drehen und einen Kick mit dem rechten Fuß auszuführen. Jede Bewegung wird mit einem farblichen Label markiert, um so einen schnellen Überblick eines Motionnets zu bekommen.

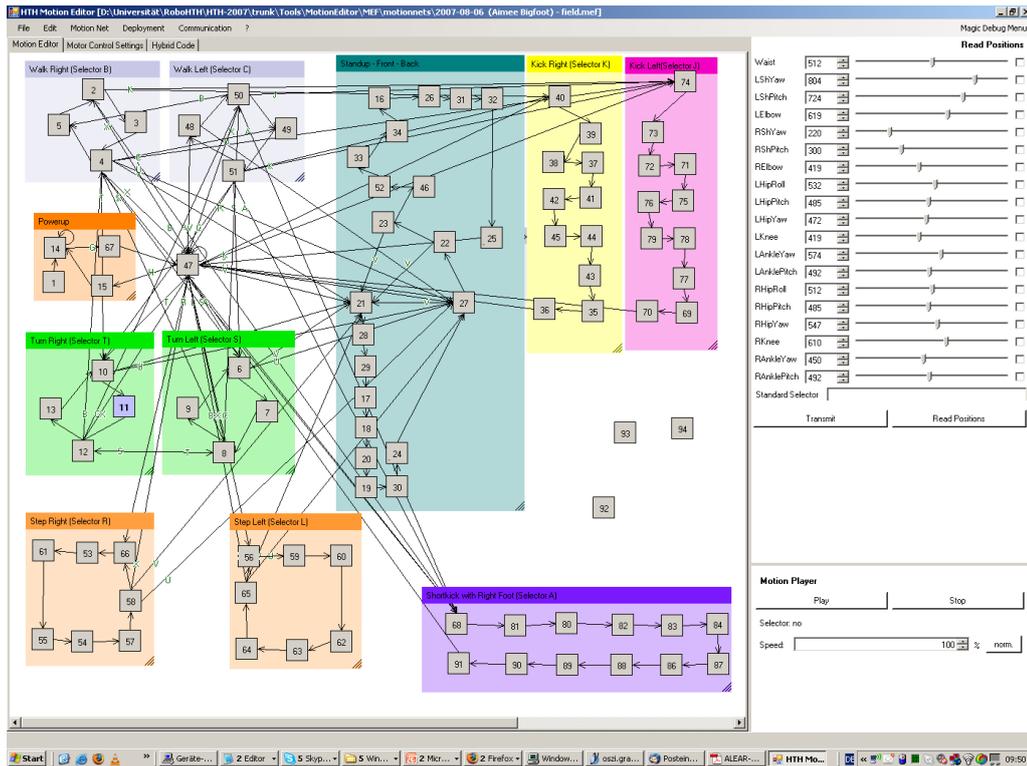


Abbildung 2.1: MotionEditor: Im rechten Drittel sind die 19 DOFs zu sehen. Auf der linken Seite ist ein Motionnet abgebildet, das folgende Bewegungen ermöglicht: Türkis: Links und Rechts gehen, Hellgrün : Auf der Stelle links und rechts drehen, Orange: Seitwärtsschritt links und rechts, Mintgrün: Aufstehen, Gelb: Mit dem rechten Fuß kicken, Rot: Langsames Anfahren der Standposition nach dem Einschalten.

2.1.1 Diskussion der Keyframe-Technik

Ein großer Vorteil der Keyframe-Technik ist die intuitive Erstellung von Bewegungen und die Erzeugung von sowohl zyklischen als auch nicht zyklischen Bewegungsmustern. Andere Playback-Verfahren berechnen die Trajektorien mit Hilfe mathematischer Modelle. Mag dies bei Bewegungspattern wie Gehen noch einfach umsetzbar sein, wird es um einiges komplizierter, wenn der Roboter komplexe Bewegungsabläufe, wie etwa das Aufstehen aus auf dem Rücken liegender Position, ausführen soll. Hierbei müssen sehr viele Zwangsbedingungen, zum Beispiel der Bewegungsspielraum der Gliedmasen oder das maximale Drehmoment, welches die Motoren aufbringen können, berücksichtigt werden. Wird dagegen mit der Unterstützung des MotionEditors eine Bewegung erzeugt, so fließen diese Einschränkungen bei der Erstellung mit ein. Weiterhin ist es möglich, jedes beliebige Bewegungspattern in eine Abfolge von Keyframes zu zerlegen [Hei07]. Die Keyframe-Technik stößt wie alle Playback-Methoden an ihre Grenzen, sobald leichte Veränderungen am Roboter

selbst oder der Umwelt auftreten. Den Roboter betreffend kann dies zur Instabilität der Bewegung führen, so dass z.B. das Aufstehen des Roboters nicht mehr gelingt. Da ein mechanisches System ständigen Veränderungen durch Abnutzung der Getriebe und sich lockernder Schrauben unterliegt, stellt dies ein sehr großes Problem dar. Ferner kann eine fest vorgegebene Bewegung nicht auf die sich verändernde Umwelt reagieren. So führen schon kleine Unebenheiten des Untergrunds zum Sturz während des Gehzyklus, der mit Hilfe der Keyframe-Technik erstellt wurde. So ist ein stetiges Anpassen des Motionnets an die sich veränderten Bedingungen des Roboters und dessen Umwelt nötig. Es besteht jedoch die Möglichkeit neuronale Controller zur Stabilisierung von Teilbewegungen im sogenannten Hybridmodus zu integrieren bzw. ganze Bewegungsmuster mit ihnen zu generieren [Thi07]. In der Dissertation von Dr. Manfred Hild wird eine Methode vorgestellt, mit der sich die Motionnets in neuronale Netze umwandeln lassen [Hil07]. Dieses Netz könnte durch künstliche Evolution optimiert werden, um stabilere Bewegungen zu erhalten.

2.2 Central-Pattern-Generator

„Central pattern generators are neuronal circuits that when activated can produce rhythmic motor patterns such as walking, breathing, flying, and swimming in the absence of sensory or descending inputs that carry specific timing information” [EM01]

2.2.1 Biologische Motivation

Diese Eigenschaft neuronaler Strukturen wurde das erste Mal in vitro von Wilson 1961 [Hoo00] nachgewiesen. Hartline identifizierte eine neuronale Struktur aus neun Zellen, die den Herzschlag des Hummers steuert [Har79]. In der Seeschnecke, *Tritonia Diomedia*, konnte die Struktur identifiziert werden, die für die Schwimmbewegung verantwortlich ist. Dies ermöglicht der Seeschnecke, Jägern zu entkommen [EM01].

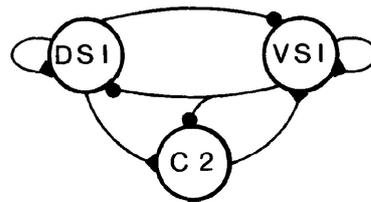


Abbildung 2.2: *Das Neuronale Netz der Seeschnecke zur Erzeugung des Schwimmpatterns enthält drei Typen von Zellen. Das dorsale Schwimminterneuron (DSI) und das ventrale Schwimminterneuron (VSI) sind die Premotorneuronen des dorsalen und ventralen Flexormuskels. Das C2 Interneuron spielt eine komplexe Rolle bei der Erzeugung des Pattern, weil es von DSI erregt und vom CSI gehemmt wird. Erregende Synapsen sind durch Dreiecke, hemmende durch Kreise dargestellt [Hoo00].*

Im Laufe der Jahre wurden immer mehr neuronale Strukturen in Lebewesen identifiziert, die als Oszillatoren fungieren. So geht man mittlerweile davon aus, dass die meisten periodischen Bewegungsabläufe von CPGs erzeugt werden. Neuronale Strukturen bestehen in der Natur aus Nervenzellen, die über Synapsen verbunden sind. Dabei gibt es zwei Typen von Synapsen: hemmende und erregende. Eine Nervenzelle (Neuron) wird solange von anderen Nervenzellen über Synapsen erregt, bis ein gewisser Schwellwert überstiegen ist. Dann fängt die Nervenzelle selbst an zu feuern. Dabei steigt die Feuerrate zu Beginn sprunghaft an und fällt dann allmählich ab. Dieser Abfall der Feuerrate wird Adaption genannt [Mat85]. In der Natur gibt es zwei Familien der CPGs. Zur ersten gehören die zellgetriebenen Oszillatoren. Hier wird die Oszillation von einem einzigen Neuron erzeugt, welches einen rhythmischen Output hat. In der zweiten Familie entsteht die Oszillation als ein Ergebnis der synaptischen Vernetzung, auch genannt Netzwerk-Oszillatoren [AS85]. Bei Netzwerk-Oszillatoren gibt es verschiedene Typen. Diese Netze unterscheiden sich in der Art der Verbindungen zwischen den Neuronen. Es gibt Netze, die ausschließlich hemmende Synapsen enthalten, weitere mit nur verstärkenden Verbindungen und jene die sowohl aus hemmenden als auch aus verstärkenden Verbindungen bestehen. Jedoch sind die meisten CPGs, die bisher entdeckt wurden, als rein hemmende Netzwerke identifiziert [AS85].

2.2.2 Neuronenmodell von Kiyotoshi Matsuoka

Kiyotoshi Matsuoka lieferte 1985 ein mathematisches Modell, welches die Eigenschaften der Nervenzellen beschreibt. Neu an diesem Modell ist die Abbildung der Adaption der Neuronen, welche als eine essentielle Eigenschaft zur Erzeugung einer schwingenden Netzwerkaktivität erkannt wurde [Rei85]. Im Folgenden wird dieses mathematische Modell eines Neurons vorgestellt, da es eins der meist verwendeten Modelle ist, die momentan zur adaptiven Steuerung von Robotern eingesetzt werden [HK07], [GE08]. Bei diesem Modell ist Zeit kontinuierlich und die Feuerrate des Neurons wird in einer kontinuierlichen Variablen kodiert.

Folgende drei Gleichungen beschreiben das Neuron:

$$\tau \dot{x} + x = \sum_{j=1}^n c_j s_j - bx' \quad (2.1)$$

$$T \dot{x} = y \quad (2.2)$$

$$y = g(x - \Theta) \quad (2.3)$$

- x Membranpotential
- τ Zeitkonstante, die den Abfall der Erregung steuert
- s_j Impulsrate der eingehenden Synapse
- c_j Gewicht der Synapse
- Θ Schwellwert, ab dem das Neuron anfängt zu feuern
- x' Grad der Adaption

$T > 0$ und $b \geq 0$ bestimmen den zeitlichen Verlauf der Adaption.

Bei diesem Modell ist darauf zu achten, dass durch richtige Einstellung eine (gedämpfte) Schwingung vermieden wird. Dies wäre ein unnatürliches Verhalten des Neurons. Die Bedingung gibt Matsuoka mit der Ungleichung $T - \tau^2 < 4T\tau b$ an [Mat85].

2.2.3 Diskussion des neuronalen Central-Pattern-Generators

Mit Hilfe dieses Neuronenmodells und der richtigen Vernetzung lassen sich nun Netze mit oszillierendem Ausgang erzeugen. Durch die Wahl der Gewichte und der Netzstruktur lassen sich die Frequenz und die Form des Ausgangssignals der Neuronen des CPGs bestimmen. Dies ist in der Abbildung 2.3 zu sehen.

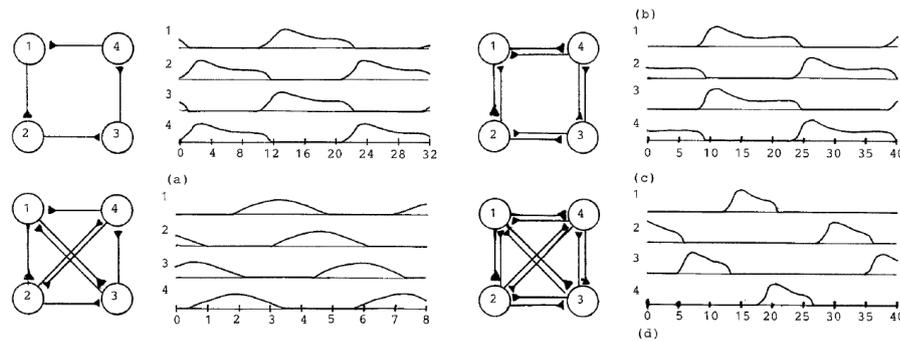


Abbildung 2.3: Neuronenringe bestehend aus vier Neuronen und ihre rhythmische Aktivität. Die Neuronen erhalten denselben Wert an erregender Aktivität von außerhalb des Netzes und dieselbe hemmende Aktivität von den internen Neuronen [Mat85].

In [Hil07] wird gezeigt, dass anhand von CPGs sich gewünschte Bewegungsmuster von kleinen Laufrobotern erzeugen lassen. Unter Verwendung eines zeitdiskreten Neuronenmodells, welches im folgenden Kapitel eingeführt wird, können neuronale Oszillatoren sehr effizient auf Microcontrollern implementiert werden.

Die durch einen CPG erzeugten Bewegungsmuster bieten jedoch keinen Vorteil gegenüber den durch Offline-Verfahren erstellten, da sie ebenfalls nicht adaptiv auf die Umwelt reagieren können. In der Biologie spricht man bei in vitro erzeugten Pattern von "fiktiven" Motorpattern. Diese Pattern können sich in der Phase und Frequenz von den in vivo erzeugten Pattern unterscheiden. Dies konnte unter anderem anhand

der Bewegungsmuster von fliegenden Heuschrecken nachgewiesen werden [EM01]. Es wurde festgestellt, dass sensorische Rückkopplung sowohl die Frequenz als auch die Phase eines CPGs beeinflusst und darüber hinaus diesen zurücksetzen kann [Hoo00]. Dies ermöglicht eine Patternerzeugung, die adaptiv auf die Umwelt reagiert. Diese sensomotorische Schleife ist in Abbildung 2.4 dargestellt.

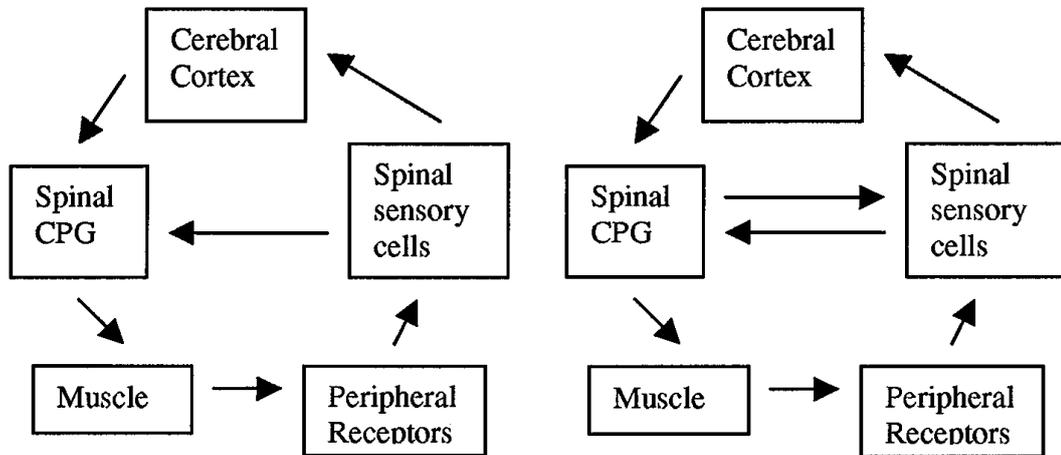


Abbildung 2.4: *Altes Modell*
beinhaltet nur unidirektionale
Sensor-Motor-Interaktion

Abbildung 2.5: *Neues Modell*
beinhaltet bidirektionale
Sensor-Motor-Interaktion

Nach neueren Untersuchungen geht das Wechselspiel zwischen den CPG-Zellen und den Sensor-Zellen über eine einseitige Verbindung zwischen Sensor und CPG hinaus. So schlägt Cohen ein Modell vor, das eine beiderseitige Beeinflussung der Zellen beinhaltet (siehe Abbildung 2.5). Dies ermöglicht zusätzlich zur Korrektur der Periodendauer und der Amplitude eines Zyklus des CPGs, eine Filterung der Sensorsignale. Somit erlaubt die bidirektionale Verbindung zwischen CPGs und Sensorzellen eine Interpretation des Sensorinputs in Abhängigkeit der Phase des Oszillators [AC99].

Arthur Kuo schlägt ein Modell zur Steuerung von rhythmischen Bewegungen anhand eines Pendels vor, in der ein CPG und Reflexschleifen zur Steuerung des Pendels eingesetzt werden. Hierbei dient der CPG als eine Art Zustandsvorhersage des Systems. Der Zustand des CPGs wird mit dem Sensorsignal verglichen, daraus der Fehler des Systems berechnet und an die Reflexschleifen weitergeleitet. Mit diesem Verfahren wurde eine Robustheit des Systems sowohl gegen Störungen von außen also auch gegen Sensorrauschen erreicht [Kuo02].

Die in den letzten Jahren gewonnenen Erkenntnisse über die neuronalen Strukturen in den Steuerungssystemen von Tieren ermöglichten eine Extraktion der Regulationsmechanismen. Dieses Wissen erlaubt nun den Bau von Controllern, die sowohl ohne Modellbildung des Robotersystems als auch der Umwelt auskommen. Diese

Controller sind hochgradig adaptiv gegenüber Veränderungen der Umwelt und am Körper. Viele regelnde Mechanismen sind lediglich im Ansatz oder noch gar nicht erfasst. Vor allem im Bereich der Sensormotorintegration ist noch vieles unklar [AC99].

Ein weiteres Problem der Neurocontroller, selbst wenn die Struktur klar sein sollte, besteht im Finden der optimalen Gewichte. Das Einstellen der Gewichte per Hand ist möglich, doch bei komplizierteren Netzen müßig bis nicht durchführbar. In der Natur wurde die neuronale Steuerung zeitgleich mit dem Körper evolviert. So ist nachgewiesen, dass die neuronalen Oszillatoren bereits vor der Geburt aktiv sind. Jedoch entspricht der Output der CPGs noch nicht dem Output eines erwachsenen Tieres [Lew02]. Allerdings finden sich in der Fauna Beispiele für eine schnelle Anpassung des CPGs. Zum Beispiel lernt ein neugeborenes Fohlen innerhalb weniger Minuten das Laufen. Auf diesen Aspekt hin hat Anthony Lewis einen Algorithmus entwickelt, der es dem vierbeinigen Roboter GEO-II wenige Minuten nach dem Einschalten ermöglicht zu gehen. In diesem Algorithmus wird der CPG als immanentes Modell der Bewegung verstanden. Der CPG wird dahingehend verändert, dass er eine immer bessere Vorhersage des Sensoroutputs macht. Weiterhin ist dieser adaptive Controller in der Lage, Gewichte, die über den Roboter verteilt werden, auszugleichen [Lew02]. Abschließend lässt sich sagen, dass neuronale Oszillatoren mittlerweile einen realisierbaren Weg darstellen, Bewegungspattern für Roboter mit Beinen zu erzeugen. So wurde eine neuronale CPG-basierende Steuerung auf den vierbeinigen Robotern Tekken 1 und 2 implementiert [HK07]. Dass diese Art der Steuerung auch für zweibeinige Fortbewegung geeignet ist, zeigt die Implementierung auf einer humanoiden Roboterplattform [GE08].

3 Neuromodule

Unter Neuromodulen werden kleine neuronale Netze verstanden, die funktionale Eigenschaften haben. In diesem Kapitel werden einige Neuromodule eingeführt und ihre spezifischen Eigenschaften näher betrachtet. Die vorgestellten Neuromodule kommen in dem beschriebenen neuronalen Netz zur Anwendung.

3.1 Neuronenmodell

In diesem Abschnitt wird das verwendete Neuronenmodell eingeführt. Dieses Modell ist ein Derivat des Modells, das von dem Neurobiologen McCulloch und dem Statistiker Pitts 1943 vorgeschlagen wurde. Hierbei wird ein diskretes Zeitmodell verwendet, da die Rechenleistung auf autonomen Robotersystemen meist sehr beschränkt ist und zeitkontinuierliche Modelle aufwendige Integrationsprozesse erfordern würden. Weiterhin werden die Signale der Neuronen Amplitudenkodiert. Dies ist gegenüber der Frequenzkodierung leichter zu implementieren und mathematisch einfacher zu beschreiben [Hil07].

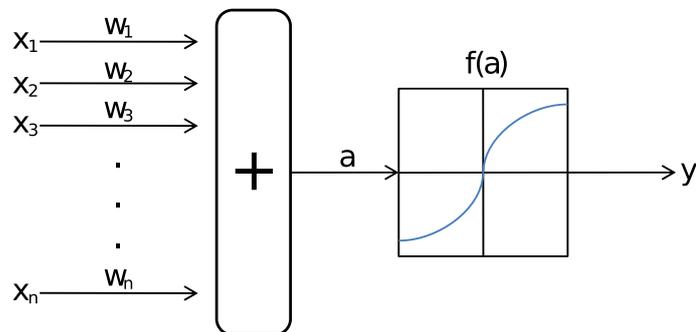


Abbildung 3.1:

Das Neuronenmodell ist in Abbildung 3.1 zu sehen. Ein Neuron besitzt n Eingänge $x_1, x_2, x_3, \dots, x_n$ und einen Ausgang y . Jeder Eingang besitzt ein Gewicht w_i . Der Wert, der am Eingang anliegt, wird mit dem Eingangsgewicht multipliziert und mit den anderen Produkten der Eingänge zur so genannten Neuronenaktivität a addiert. Danach wird das Ausgangssignal mit Hilfe der Übertragungsfunktion f aus der Aktivität a berechnet.

$$a := \sum_{i=1}^n w_i x_i \quad w_i, x_i \in \mathbb{R} \text{ für } i = 1, 2, 3, \dots, n \quad (3.1)$$

$$y := f(a) \quad (3.2)$$

Eine Vereinfachung der Schreibweise lässt sich erreichen, wenn die Eingangssignale x_i und die Gewichte w_i zu n -dimensionalen Vektoren zusammengefasst werden. Da es sich um ein zeitdiskretes Modell handelt, lässt sich nun das Ausgangssignal für jeden Zeitpunkt $t \in \mathbb{N}$ wie folgt berechnen [Hil07]:

$$a(t+1) := \langle w(t), x(t) \rangle \quad w(t), x(t) \in \mathbb{R}^n, t \in \mathbb{N} \quad (3.3)$$

$$y(t+1) := f(a(t+1)) \quad (3.4)$$

3.1.1 Transferfunktion

In der Literatur gibt es eine große Anzahl von Funktionen, die als Transferfunktion Verwendung finden. Die häufigsten sind Identität, Sprungfunktion, Standard-Sigmoide und der Tangens-Hyperbolicus. Im Folgenden wird ausschließlich der Tangens-Hyperbolicus verwendet, da dieser einige vorteilhafte Eigenschaften besitzt. Durch diese Eigenschaften lassen sich die anderen oben genannten Funktionen einfach approximieren. Weiterhin ist der Tangens-Hyperbolicus auf ganz \mathbb{R} definiert, stetig differenzierbar, streng monoton steigend, umkehrbar und auf den Bereich $] -1; 1[$ beschränkt [Hil07].

Der Kurvenverlauf des Tangens-Hyperbolicus ist in Abbildung 3.2 zu sehen. Dort

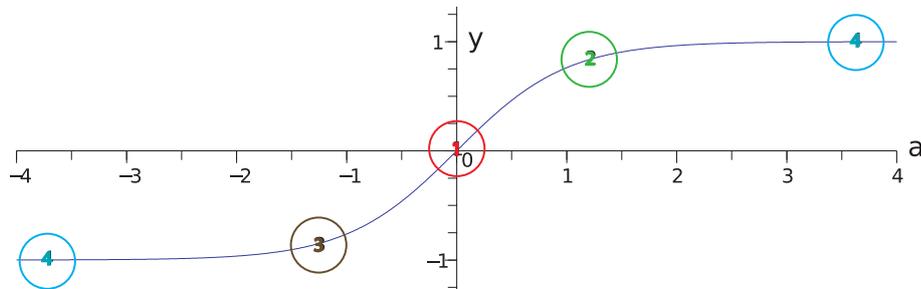


Abbildung 3.2: Tangens-Hyperbolicus mit markierten Arbeitsbereichen.

sind die einzelnen Arbeitsbereiche markiert. So lässt sich mithilfe der Gewichte und des Bias-Wertes sehr leicht eine Verschiebung in den gewünschten Arbeitsbereich erreichen.

- 1. Linearer Arbeitsbereich:** Kleine Eingangsgewichte bewirken, dass sich die Aktivität nur in dem Bereich bewegt, in dem der Tangens-Hyperbolicus annähernd

linear ist. Daher lassen sich LIR-Systeme mit Hilfe von Neuronen approximieren. Dies hat zur Folge, dass sich Neuronen sehr gut zu FIR- und IIR-Filtern verschalten lassen.

2. **Logarithmisch:** Mit Hilfe dieses Arbeitsbereiches und dem exponentiellen Bereich ist es möglich Multiplikation auszuführen.

$$a \cdot b = e^{\log(a)+\log(b)} \quad (3.5)$$

3. **Exponentiell:** Dieser Bereich kann verwendet werden um nicht lineare Eigenschaften zu erzielen
4. **Sättigung:** Durch hohe Eingangsgewichte nähert sich der Tangens-Hyperbolicus der Sprungfunktion an. Somit fungiert das Neuron als Schalter mit den Werten von annähernd -1 und 1. Auf diese Weise kann man Neuronen auch für boolesche Algebra verwenden.

3.1.2 Neuronales Netz

Ein Neuronales Netz besteht aus einer Menge von Neuronen die untereinander verknüpft sind. Es gibt in einem Netz drei Typen von Neuronen:

Inputneuron: Dies sind Neuronen, die Sensorwerte zur Verfügung stellen. In der A-Serie sind dies Beschleunigungs- und Winkelwerte. Weiterhin sind Inputneuronen implementiert mit deren Hilfe von außen Signale in das Netz geleitet werden können, zum Beispiel von Menschen über die Tastatur oder von höheren kognitiven Schichten.

Outputneuron: Das Ausgangssignal dieser Neuronen steuert Aktuatoren und LEDs der A-Serie.

Hiddenneuron: Dies sind alle restlichen Neuronen. Sie dienen der Verarbeitung der Input-Werte zu Output-Werten.

Mit Hilfe der Wahl des Gewichtes der Kante zwischen zwei Neuronen lässt sich die Eigenschaft der Verbindung bestimmen.

- $|w| < 1$ Abschwächung
- $|w| > 1$ Verstärkung
- $w < 0$ Phasenumkehr
- $w = 0$ keine Verbindung

3.1.3 Biasneuron

Da es nötig sein kann, Neuronen mit einer Grundaktivität zu versehen, wird ein spezielles Neuron definiert: das Biasneuron. Dies besitzt die Eigenschaft immer einen Output-Wert von 1 zu liefern. Durch die Verbindung zu diesem Neuron lassen sich nun andere Neuronen mit einer Aktivität vorspannen. Dies kann nötig sein, um Offset-Werte von Sensoren auszugleichen oder den Arbeitsbereich an die gewünschte Position zu schieben [Hil07].

3.1.4 Matrix-Schreibweise eines neuronalen Netzes

Ein neuronales Netz lässt sich sehr kompakt als Gewichtsmatrix $\mathbf{W} \in \mathbb{R}^{(n_o+n_h) \times (n_o+n_h+n_i)}$ darstellen. Hierbei ist n_o die Anzahl der Outputneuronen, n_i der Inputneuronen und n_h der Hiddenneuronen. Der Eintrag w_{ij} steht für das Gewicht der Kante von dem Ausgang des Neurons j zu dem Eingang des Neurons i . Das Biasneuron wäre in dieser Darstellung eines der Inputneuronen. Das Netz lässt sich nun wie folgt berechnen:

$$\hat{\mathbf{x}}(t+1) := \tanh(\mathbf{W} \cdot \mathbf{x}(t)) \quad t \in \mathbb{N} \quad (3.6)$$

$$\hat{\mathbf{x}} := \begin{pmatrix} x_1 \\ \vdots \\ x_{n_o} \\ \vdots \\ x_{n_o+n_h} \end{pmatrix} \quad (3.7)$$

$$\mathbf{x} := \begin{pmatrix} x_1 \\ \vdots \\ x_{n_i} \\ \hat{\mathbf{x}} \end{pmatrix} \quad (3.8)$$

Der Tangens-Hyperbolicus wird hierbei elementweise auf die einzelnen Elemente des Aktivitätsvektors angewandt [Hil07].

3.1.5 Vereinfachte Darstellung

Da die Darstellung eines Neurons in Abbildung 3.1 sehr groß ist, werde ich eine vereinfachte Symbolik wie in Abbildung 3.3 verwenden. Hierbei wird das Gewicht der Kante zum Bias-Neuron in das Neuron geschrieben. Die Gewichte aller anderen Verbindungen werden weiterhin an die Kante geschrieben. Der Übergang von Aktivität und Transferfunktion wird als ein Kreis zusammengefasst.

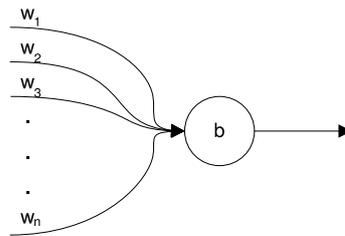


Abbildung 3.3: Vereinfachte Darstellung eines Neurons

3.2 Neuromodule mit einem Neuron

Mit Hilfe eines Neurons in Verbindung mit und ohne Selbstkopplung lassen sich wichtige Grundbausteine der Signalverarbeitung erzeugen.

3.2.1 Schwellwertschalter

Ein großes Eingangsgewicht bewirkt, dass sich der Tangens-Hyperbolicus einer Sprungfunktion annähert. Anhand des Bias-Wertes lässt sich der Nulldurchgang nach links oder rechts verschieben. In der Abbildung 3.4 wurde ein Eingangsgewicht von acht gewählt, da dies der höchste zulässige Wert im neuronalen Bytecode ist, der in der Firmware der A-Serie implementiert ist [Thi07].

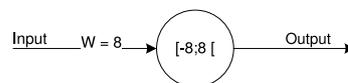


Abbildung 3.4: Darstellung eines neuronalen Schwellwertschalters

Sollte die Steilheit des Tangens-Hyperbolicus nicht ausreichen, können mehrere Schwellwertschalter hintereinander geschaltet werden, um einen steileren Kurvenverlauf zu erreichen. Jedoch hat sich bisher in der Anwendung gezeigt, dass eine Acht als Eingangsgewicht ausreichend ist.

3.2.2 Neuronale Filterung

Wie oben bereits erwähnt, lassen sich mit Hilfe von Neuronen lineare Systeme und damit verschiedene Filter realisieren. Um die Theorie der linearen Systeme auf neuronale Netze anwenden zu können, muss darauf geachtet werden, dass die Signalamplitude den annähernd linearen Bereich des Tangens-Hyperbolicus nicht verlässt. Sollte diese Regel verletzt werden, treten zusätzlich nicht lineare Verzerrungen auf [Hil07]. Mittels eines Neurons mit Selbstkopplung lässt sich ein IIR-Filter erster

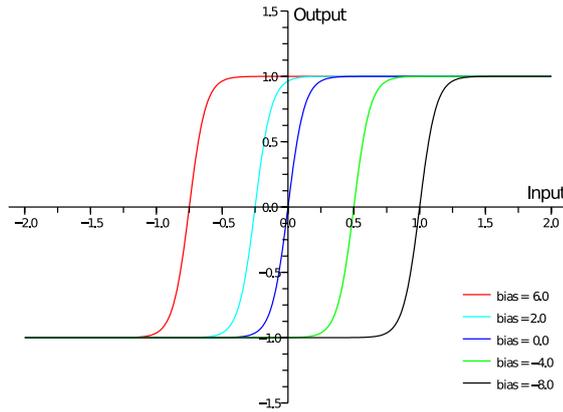


Abbildung 3.5: *Input/Output-Verhalten eines Schwellwertschalters mit verschiedenen Bias-Werten*

Ordnung realisieren. Es kann sowohl als Tiefpass- als auch als Hochpassfilter funktionieren.

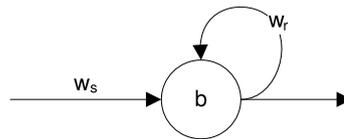


Abbildung 3.6: *Neuron mit Eingangsgewicht w_s und einer Selbstkopplung w_r*

Der Ausgang des Neurons berechnet sich dann wie folgt:

$$y(t+1) = \tanh(w_s \cdot x(t) + w_r \cdot y(t)) \quad (3.9)$$

Für kleine Signalamplituden gilt:

$$y(t+1) \approx w_s \cdot x(t) + w_r \cdot y(t) \quad (3.10)$$

Entscheidend, ob eine Tiefpassfilterung oder eine Hochpassfilterung des Signals stattfindet, ist das Vorzeichen des Gewichtes der Selbstkopplung. Frequenz- und Phasengang des Filterneurons lassen sich durch die Z-Transformation bestimmen.

$$H(z) = \frac{w_s \cdot z^0}{1 - w_r \cdot z^{-1}} \quad (3.11)$$

Der Amplitudengang berechnet sich dann wie folgt:

$$A(f) = \left| \frac{w_s}{1 - w_r \cdot e^{-i \cot 2\pi \cdot \frac{f}{f_s}}} \right| \quad (3.12)$$

Hier ist f eine Frequenzkomponente des Eingangssignals und f_s ist die Updaterate des Neurons. Dies entspricht somit der Abtastfrequenz des Signals.

Der Phasengang berechnet sich folgendermaßen:

$$\phi(f) = \left(\frac{\operatorname{Im} \left(\frac{w_s}{1 - w_r \cdot e^{-i \cot 2\pi \cdot \frac{f}{f_s}}} \right)}{\operatorname{Re} \left(\frac{w_s}{1 - w_r \cdot e^{-i \cot 2\pi \cdot \frac{f}{f_s}}} \right)} \right) \quad (3.13)$$

Tiefpasscharakteristik hat das Neuron bei einer Selbstkopplung im Bereich $w_s \in]0; 1[$. Ist die Selbstkopplung aus dem Intervall $w_s \in]-1; 0[$ hat es Hochpass-Eigenschaften. In den Abbildungen 3.7 und 3.8 sind Frequenz und Phasengänge für die jeweiligen Parametersätze zu sehen.

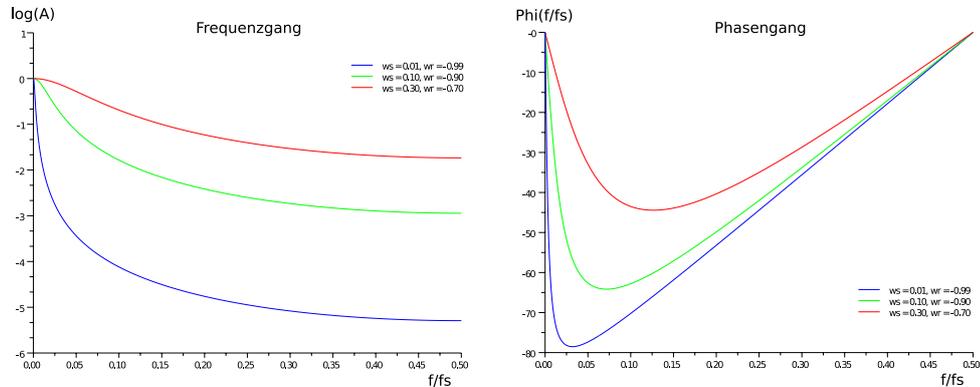


Abbildung 3.7: Frequenz- und Phasengang für verschiedene Parametersätze mit Tiefpasscharakteristik

IIR-Filter besitzen theoretisch eine unendliche Impulsantwort, jedoch fällt der Wert irgendwann soweit ab, dass er im Systemrauschen untergeht. Baut man größere Filterstrukturen mit IIR-Filtern auf, muss die Rechengenauigkeit des Gesamtsystems betrachtet werden, da es sonst durch Rechenfehler zu ungewollter Eigenschwingung des Systems kommen kann [Hil07].

Aus mehreren in Reihe geschalteten Neuronen können auch FIR-Filter aufgebaut werden. Jedes Neuron stellt eine Zeitverzögerung um einen Schritt dar. Werden nun die Ausgänge gewichtet aufaddiert, können so FIR-Filter gewünschter Charakteristik und Ordnung realisiert werden (siehe Abbildung 3.10). Dabei ist zu beachten, dass der Tangens-Hyperbolicus nur annähernd linear ist und es zu einer Abschwächung des Signals bei mehreren in Reihe geschalteten Neuronen kommt. Dies kann mit Verbindungsgewichten größer Eins kompensiert werden.

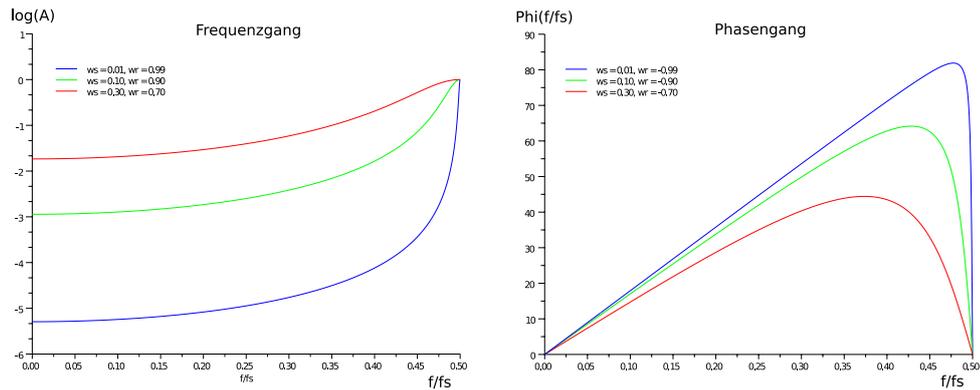


Abbildung 3.8: Frequenz- und Phasengang für verschiedene Parametersätze mit Hochpasscharakteristik

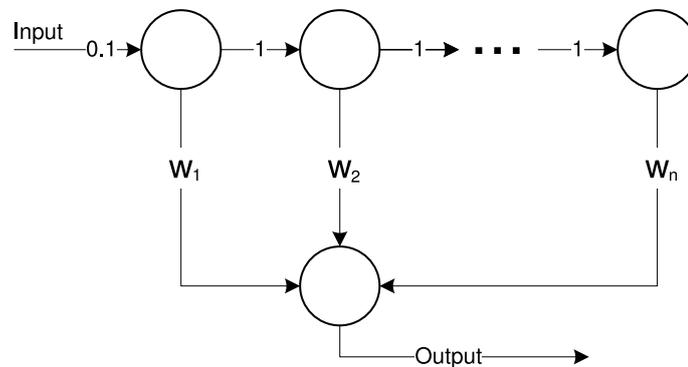


Abbildung 3.9: Neuronaler FIR-Filter n -ter Ordnung

3.2.3 Hysterese-Neuron

Bei einer Selbstkopplung $w_s > 1$ verändert sich das Verhalten des Neurons. Der ehemals stabile Fixpunkt wird instabil und von zwei stabilen Fixpunkten eingeschlossen. Es entsteht eine so genannte Hysterese. Hierbei ist der Eingangswert in dem das Neuron von einem zum anderen Fixpunkt wechselt, abhängig von dem aktuellen Zustand.

Somit kann ein Neuron als Zweipunktregler fungieren. Dieses Verhalten lässt sich verwenden, um einen Zustand $-1/1$ zu speichern.

3.3 Neuro-Module mit mehreren Neuronen

In diesem Abschnitt werden weitere Neuro-Module mit mehr als einem Neuron vorgestellt.

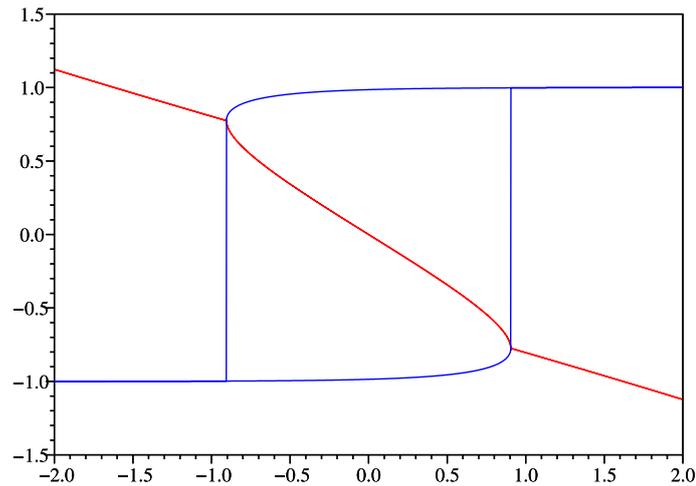


Abbildung 3.10: *Hystereseverhalten (blau) des Neurons bei einer Selbstkopplung von $w = 2,5$. Lässt man die Zeit in der Simulation rückwärts laufen, so erhält man die instabilen Fixpunkte (rot). [Hil07]*

3.3.1 Boolesche Operatoren

Wie im Abschnitt 3.1.1(Transferfunktion) erwähnt, können mit Hilfe der Neuronen boolesche Berechnungen ausgeführt werden. Verwendet man 1 für TRUE und -1 für FALSE, wird eine Vereinfachung der Schaltung erreicht, da der Sättigungsbereich des Tanges-Hyperbolicus 1 und -1 ist. Somit ist für alle booleschen Berechnungen ein hohes Eingangsgewicht Voraussetzung.

Im Folgenden betragen alle Eingangsgewichte 5.

NICHT-Schaltung: Ist sehr leicht durch ein negatives Eingangsgewicht an einem Neuron zu erreichen.

UND-Schaltung: Das Neuron darf erst 1 liefern, wenn an allen Eingängen eine 1 anliegt. Daher wird der Bias wie folgt gewählt: Bei n-Eingängen ergibt sich ein Bias von $-1*(n-1)*5$.

ODER-Schaltung: Das Neuron soll 1 liefern, sobald ein Eingang nicht -1 ist. Daraus folgt eine Bildungsvorschrift für den Bias $(n-1)*5$.

XOR-Schaltung: Für eine XOR-Schaltung gibt es verschiedene Möglichkeiten. Die einfachste und auf beliebig viele Eingänge erweiterbare Schaltung setzt sich zusammen aus n-UND- Schaltungen mit einer nachgeschalteten n-OR-Schaltung. Für nur zwei Eingänge lässt sich eine Schaltung mit zwei Neuronen erstellen (siehe Abbildung 3.11). Bei dieser Schaltung wird der Fall, dass beide Eingänge TRUE liefern, vom OR-Fall abgezogen. Somit liefert diese Schaltung ein XOR-Verhalten.

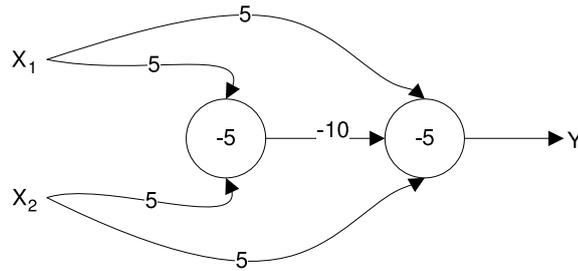


Abbildung 3.11: XOR-Schaltung mit zwei Neuronen

3.3.2 Gate-Schaltung

Innerhalb eines neuronalen Netzes kann es nötig sein gewisse Netzbereiche vom Signalfluss abzutrennen oder hinzu zu schalten, um eine gewünschte Verhaltensänderung des Netzes zu erreichen. Der Modulaufbau ist in Abbildung 3.12 zu sehen.

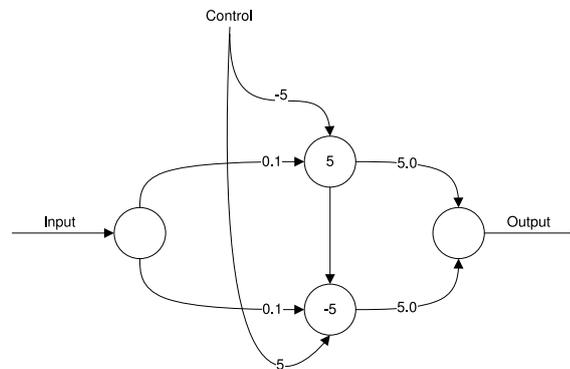


Abbildung 3.12: Gate-Schaltung mit einer Input-Leitung und einer Steuerleitung(Control)

Liegt nun eine -1 am Controlleingang an, ist das Gate gesperrt. Bei einer 1 liegt der Input am Ausgang des Gates an. Bei dieser Schaltung wird zum einen der lineare Arbeitsbereich als auch die Sättigung des Tangens-Hyperbolicus verwendet. Bei $p = 1$ werden die großen Bias-Werte der Gate-Neuronen aufgehoben und das Signal gelangt durch die kleinen Eingangsgewichte nahezu unverzerrt durch das Gate. Bei $p = -1$ wird das Signal durch die hohen Steuergewichte und den Bias in die Sättigung gezogen. Die durch die Eingangsgewichte kleine Signalamplitude ist dort nicht mehr bemerkbar. So ergibt sich bei der Addition der Ausgänge der Gate-Neuronen ein Wert von 0 . Betrachtet man nun den Signalverlauf für verschiedene Steuerparameter ergibt sich folgendes Diagramm in Abbildung 3.13.

Hier ist zu sehen, dass im Wertebereich zwischen $]0; 1[$ das Signal nicht ganz verschwindet, sondern nur eine Abschwächung stattfindet. Dies ist einerseits wichtig,

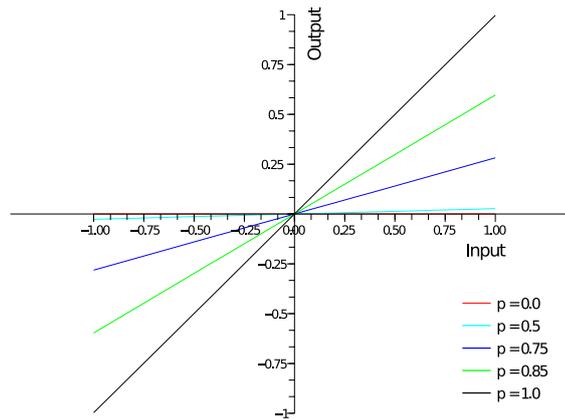


Abbildung 3.13: *Input/Output-Verhalten des Gates bei verschiedenen Steuerparametern*

da darauf geachtet werden muss, dass das Steuersignal sich nicht in den Bereichen aufhält, die von einer Abschwächung betroffen sind. Andererseits führt dies zum nächsten Neuromodul, in dem dieses Verhalten gezielt ausgenutzt wird.

3.3.3 Linearer Multiplikator

Gerade bei der Bewegungssteuerung kann es sehr interessant sein, Bereiche von Signalen nicht komplett abzuschalten, sondern nur abzuschwächen, um eine Bewegung mit kleiner Amplitude auszuführen. Wie zuvor bereits erwähnt, zählt dies bereits zu den Eigenschaften der Gate-Schaltung. Jedoch findet diese Abschwächung mit einem nichtlinearen Faktor statt. Das erschwert die manuelle Konstruktion eines Netzes, da es für Menschen leichter ist, mit linearen Faktoren zu rechnen. Auch ist es mit einem größeren Wertebereich zwischen Auslöschung und Durchlass leichter das gewünschte Verhalten einzustellen.

Im Wesentlichen kann dieses Modul in zwei Bereiche aufgeteilt werden. Zum einen ist eine Gate-Schaltung mit veränderten Gewichten zu erkennen (siehe Abbildung 3.15). Durch die kleineren Gewichte ergibt sich ein veränderter Signalverlauf für die Gate-Struktur. Das Signal wird nun erst bei -1 nahezu ausgelöscht und ist bei $p = 0$ schon deutlich größer als 0.

Das Verhältnis von Steuersignal zur Übertragungsfunktion ist dennoch noch nicht linear. Die Übertragungsfunktion des linearen Multiplikators lässt sich wie folgt darstellen.

$$Output(x) = f(p) \cdot Input \quad f(p) \text{ ist nicht linear} \quad (3.14)$$

Der zweite Teil des Netzes approximiert $f(p)^{-1}$

Die Gewichte wurden per Hand eingestellt. Die gebildete Funktion entspricht einer verzerrten und verschobenen Umkehrfunktion des Tangens-Hyperbolicus.

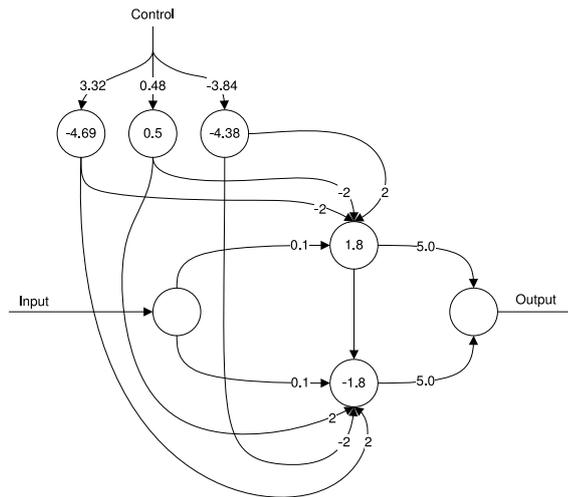


Abbildung 3.14: *Linearer Multiplikator mit einem Input-Eingang und einem Steuereingang(Control)*

$$Output(x) \approx \frac{(p+1)}{2} \cdot Input \quad (3.15)$$

Somit ergibt sich für den linearen Multiplikator für verschiedene Steuerparameter folgender Signalverlauf (Abbildung 3.17).

In der Abbildung 3.17 ist gut zu erkennen, dass das Signal nicht ganz null wird. Der maximale Fehler bei $p = -1$ beträgt für ein Eingangssignal von 1 ca. 0.015. In folgender Anwendung ist dieser Fehler zu vernachlässigen. Die verwendeten Motoren in der A-Serie besitzen eine Winkelauflösung von 0.35° bei 300° Aktionsradius. Somit entspricht ein Grad einem Wert von ca. 0.0067. Da der Fehler bei sinkender Amplitude linear kleiner wird, ist er bei einem Input von 0.5 nur noch 0.0075. Der Fehler fällt sehr schnell unter einen Wert, auf den der Motor reagiert.

3.3.4 Neuronale Oszillatoren

In Kapitel 2.2 „Central Pattern Generator“ wurden bereits einige Strukturen, die als Oszillator fungieren, vorgestellt. Sie standen allerdings in Verbindung mit dem Neuronenmodell von Matsuoka. In diesem Abschnitt wird ein Oszillator mit zwei Neuronen vorgestellt unter der Verwendung des zeitdiskreten Neuronenmodells aus Abschnitt 3.1. Mit zwei Neuronen lässt sich auf vielfältige Art und Weise ein oszillierendes Netzwerk aufbauen. Dabei unterscheiden sich die Oszillatoren in markanten Eigenschaften, wie der Form des Outputs, Frequenz und Amplitude [Hil07].

Die Gewichtsmatrix eines vollständigen Netzes aus zwei Neuronen besteht aus einer Matrix.

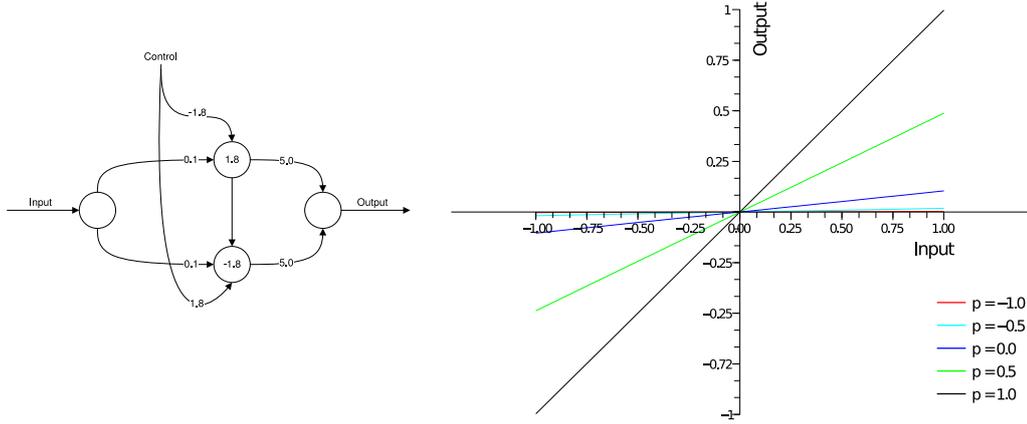


Abbildung 3.15: Links das Teilnetz mit Gate-Struktur, rechts der Signalverlauf für verschiedene Steuerparameter

$$W = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix}, W \in \mathbb{R}^{2 \times 2} \quad (3.16)$$

Die Eigenschaften eines schwingbaren Netzes lassen sich aus den Stabilitätseigenschaften der Fixpunkte des Systems herleiten. Hierbei werden die Eigenwerte der Jacobi-Matrix betrachtet. Diese ist für den Fixpunkt gegeben durch:

$$Df_w(\mathbf{x}^*) = \begin{pmatrix} w_{11} \tanh' x_1^* & w_{12} \tanh' x_2^* \\ w_{21} \tanh' x_1^* & w_{22} \tanh' x_2^* \end{pmatrix}, \quad (3.17)$$

Die Eigenwerte, $\lambda_{1,2}(\mathbf{x}^*)$ berechnen sich nun aus der Nullstellenbestimmung des charakteristischen Polynoms der Jacobi-Matrix.

$$|Df_w(\mathbf{x}^*) - \lambda_{1,2}(\mathbf{x}^*) I| \stackrel{!}{=} 0 \quad (3.18)$$

$$(w_{11} \tanh' x_1^* - \lambda(\mathbf{x}^*)) (w_{22} \tanh' x_2^* - \lambda(\mathbf{x}^*)) = 0 \quad (3.19)$$

$$- (w_{12} \tanh' x_2^*) (w_{21} \tanh' x_1^*) \quad (3.20)$$

$$\lambda(\mathbf{x}^*)^2 - (w_{11} \tanh' x_1^* + w_{22} \tanh' x_2^*) \lambda(\mathbf{x}^*) \quad (3.21)$$

$$+ (w_{11} \tanh' x_1^*) (w_{22} \tanh' x_2^*) - (w_{12} \tanh' x_2^*) (w_{21} \tanh' x_1^*) = 0 \quad (3.22)$$

Gilt für beide Eigenwerte $-1 < \lambda_{1,2}(\mathbf{x}^*) < 1$ so ist \mathbf{x}^* ein asymptotisch stabiler Fixpunkt. Ist nur ein Eigenwert betragsmäßig größer als 1 ist der Fixpunkt instabil. Bei dem Grenzfall $\lambda_{1,2}(\mathbf{x}^*) = 1$ sind höhere Ableitungen zu betrachten. Sind nun beide Eigenwerte komplex konjugiert, werden die im Fixpunkt ansetzenden Eigenvektoren um ϕ gedreht und um r gestreckt. Eine stabile Schwingung kommt durch

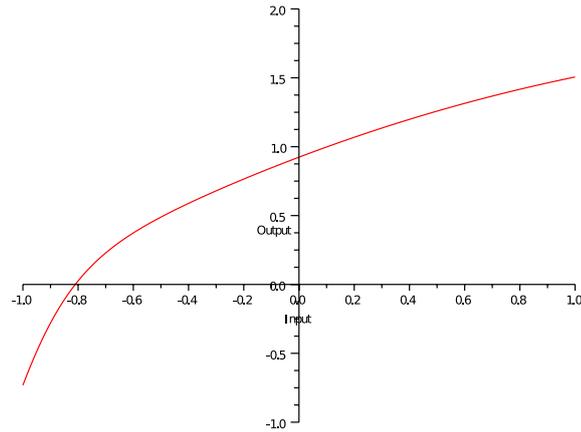


Abbildung 3.16: Übertragungsfunktion des Steuerteilnetzes des linearen Multiplikators

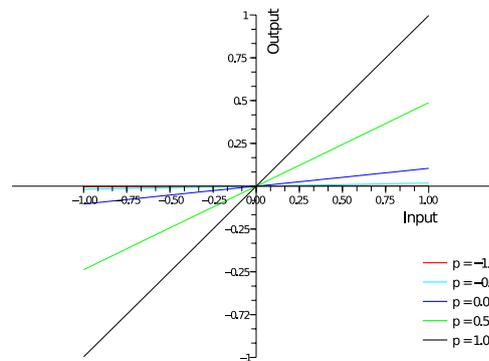


Abbildung 3.17: linearer Signalverlauf für verschiedene Steuerparameter p

die Dämpfung des Tangens Hyperbolicus nur dann zustande, wenn $r > 1$ ist [Hil07]. ϕ und r berechnen sich aus den komplexen Eigenwerten wie folgt:

$$\lambda_{1,2}(\mathbf{x}^*) = a \pm ib \quad (3.23)$$

Die kartesische Darstellung ist äquivalent in der Polarkoordinatendarstellung zu

$$\lambda_{1,2}(\mathbf{x}^*) = r e^{\pm i\phi} \quad (3.24)$$

$$r = \sqrt{a^2 + b^2} \quad (3.25)$$

$$= \sqrt{\lambda_1(x^*) \cdot \lambda_2(x^*)} \quad (3.26)$$

$$= (w_{11} \tanh' x_1^*) (w_{22} \tanh' x_1^*) - (w_{12} \tanh' x_2^*) (w_{21} \tanh' x_1^*) \quad (3.27)$$

$$\phi = \operatorname{atanh} \frac{a}{b} \quad (3.28)$$

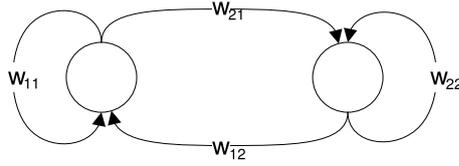


Abbildung 3.18: *Vollständig verbundenes Zwei-Neuronennetzwerk*

Damit die Eigenwerte komplex konjugiert sind, muss die Diskriminante der Gleichung 1 kleiner 0 sein. Daraus ergibt sich folgende Bedingung:

$$\frac{1}{4}(w_{11}\tanh'x_1^* + w_{22}\tanh'x_1^*) < (w_{11}\tanh'x_1^*)(w_{22}\tanh'x_1^*) \quad (3.29)$$

$$- (w_{12}\tanh'x_2^*)(w_{21}\tanh'x_1^*) \quad (3.30)$$

Sind beide Voraussetzungen erfüllt, liegt ein Oszillator vor.

3.3.5 Neuronaler $SO(2)$ -Oszillator

Im Folgenden wird näher auf den neuronalen $SO(2)$ -Oszillator eingegangen. Pasmann stellte diesen das erste Mal als neuronales Netz vor [FP03]. Es besitzt die nützliche Eigenschaft, zwei um $\frac{\pi}{2}$ verschobene Sinusschwingungen zu erzeugen. Dieser sinusförmige Output ist gut geeignet um zyklische Bewegungsmuster zu erzeugen, wie etwa ein Gehen bei Laufmaschinen. Dies wurde in [Hil07] realisiert. Weiterhin kann aus diesen zwei Schwingungen sehr leicht eine um ein beliebiges Φ phasenverschobene Schwingung erzeugt werden. Siehe hierzu Kapitel 4.2.3. .

Die Bildungsvorschrift für die Gewichtsmatrix ist wie folgt:

$$W := (1 + \epsilon) \begin{pmatrix} s & r \\ -r & s \end{pmatrix} \quad (3.31)$$

$$s := \cos(2\pi\phi) \quad (3.32)$$

$$r := \sin(2\pi\phi) \quad (3.33)$$

Es gilt $0 \leq \phi \leq 0.5$ und $0 < \epsilon \ll 1$. Die Frequenz des Oszillators hängt im Wesentlichen von ϕ und nur in geringem Maße von ϵ ab [Hil07]. Die Matrix lässt sich als Rotationsabbildung interpretieren, wobei der Faktor $(1 + \epsilon)$ dazu dient, die Dämpfung des Tangens-Hyperbolicus auszugleichen. Bei einem großen ϵ steigt die Amplitude stark an (siehe Abbildung 3.19). Hier kommt die Nichtlinearität des Tangens Hyperbolicus zum Tragen. Die sinusförmige Schwingung nähert sich einer Rechteckschwingung an.

In [Hil07] wird der neuronale $SO(2)$ zu einem Ringoszillator erweitert. Hierdurch entstehen Ringmodule, die aneinander gekoppelt werden können. Dabei bleibt die

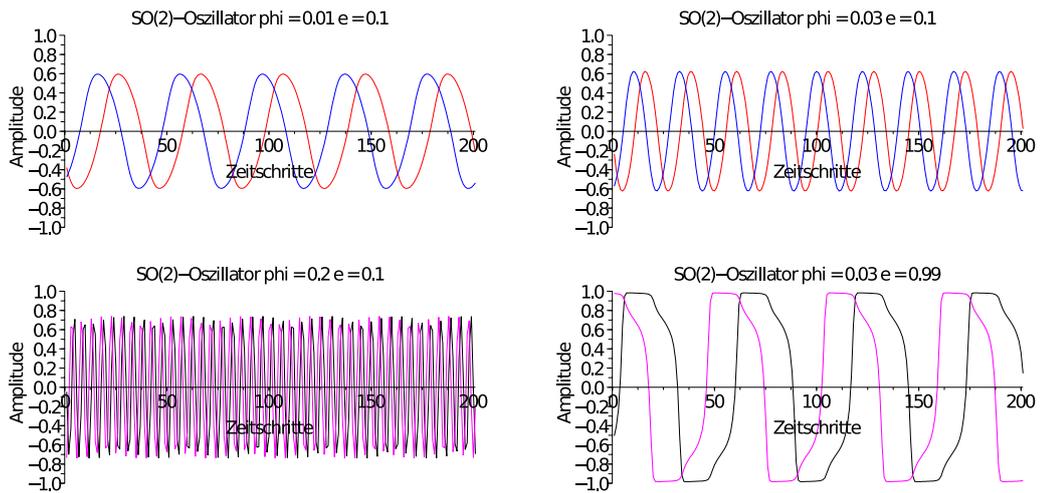


Abbildung 3.19: Output der zwei Neuronen eines $SO(2)$ -Oszillators mit verschiedenen Werten für ϵ und ϕ über 200 Zeitschritte.

Oszillation erhalten, verändert sich aber in Frequenz und Amplitude. Mit Hilfe einer homöostatischen Lernregel wird dabei eine konstante Frequenz und Amplitude erreicht. Diese Lernregel macht sich zunutze, dass die Frequenz hauptsächlich über die Ringkopplungen gesteuert werden kann. Die Amplitude ist maßgeblich von der Selbstkopplung abhängig. Diese Erkenntnis könnte dazu verwendet werden Bewegungsmuster zu erzeugen, die adaptiv auf die Umwelt reagieren.

4 Konstruktion des neuronalen Netzes

In diesem Kapitel wird das neuronale Netz vorgestellt, welches dem Roboter das Gehen in verschiedenen Kurven erlaubt. Als erstes wird auf die sensomotorische Schleife eingegangen, die die Grundschiwingung erzeugt. Der zweite Abschnitt behandelt die Topologie des kompletten Netzes, speziell die Art und Weise der Manipulierbarkeit der Grundschiwingung, die zur Gehbewegung des Roboters führt. Das Netz verfügt über fünf Steuerparameter, die folgende Aktionen erlauben:

- Start und Stopp der lateralen Schwingung
- Start-Impuls
- Start und Stopp des Ganges
- Links- oder Rechtskurve
- Größe des Kurvenradius

4.1 Neuronale Reflexschleife zur Erzeugung der Grundschiwingung

Gehen lässt sich leicht nachvollziehbar in zwei Teilbewegungen zerlegen. Zum einen in die Gewichtsverlagerung von einem Bein auf das andere, zum anderen in die Vorwärtsbewegung der Beine. Es liegt auf der Hand, dass die Gewichtsverlagerung in der Lateralebene die Grundlage für eine Beinbewegung in der Sagittalebene bildet.

4.1.1 Konstruktion der neuronalen Reflexschleife

Diese Pendelbewegung wird von einer neuronal umgesetzten sensomotorischen Schleife erzeugt. Eine sehr einfache Struktur, die eine stabile Pendelbewegung erzeugt, ist in Abbildung 4.1 zu sehen. Zuerst wird der Mittelwert der Beschleunigungsachsen der AccelBoards auf den Schultern in der lateralen Ebene gebildet. Dies entspricht der Beschleunigung der Achse durch die Körpermitte. Dieser Wert wird als Zielwinkel gewichtet auf die Motoren der Rollgelenke von Hüfte und Knöchel gegeben. Der Roboter bildet zusammen mit der neuronalen Steuerung ein dynamisches System, welches im Wesentlichen drei stabile Fixpunkte besitzt. Diese sind Stehen, auf dem

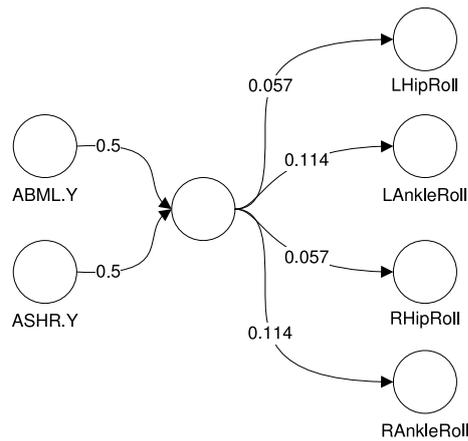


Abbildung 4.1: Einfache neuronale Reflexschleife zur Erzeugung einer lateralen Pendelbewegung

Boden liegen und der quasiperiodische Attraktor der Pendelbewegung. Durch einen leichten Stoß in der lateralen Ebene fängt der Roboter an zu pendeln. Wird der Roboter festgehalten, fällt das System wieder zurück in den Fixpunkt des Standes. In Abbildung 4.2 ist der Verlauf der Beschleunigungswerte aufgezeichnet. Diese

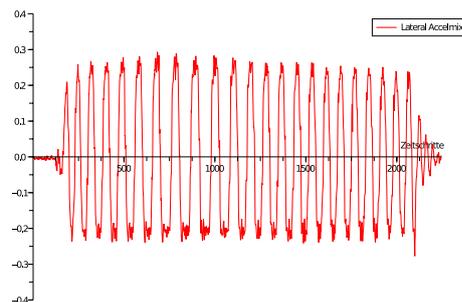


Abbildung 4.2: Mittelwert der lateralen Beschleunigungswerte mehrerer Bewegungszyklen. Start der Bewegung erfolgte durch einen leichten Stoß, Stopp durch leichtes Festhalten des Roboters

einfache Reflexschleife hat einen Nachteil. Das Signal der Beschleunigungswerte besitzt sehr viele verschiedene Frequenzanteile. Werden die hochfrequenten Anteile auf die Motoren gegeben, führt dies zu unnötigem Stromverbrauch. Daher wird, wie in Abbildung 4.3 zu sehen, die Reflexschleife um zwei IIR-Filter erster Ordnung erweitert. Das Signal wird durch jedes der Filter-Neuronen um ca. 45° Phasen verschoben. Aus den Ausgängen der zwei Filterneuronen wird eine sinusförmige Schwingung mit der Amplitude und Phase der lateralen Beschleunigung gemischt. Das Resultat wird daraufhin auf die Motoren gegeben. In Abbildung 4.4 zeigt sich an der Motorsignalkurve, dass die Filterung erfolgreich eingesetzt wurde und die Frequenzkomponenten in der Motoransteuerung stark reduziert wurden.

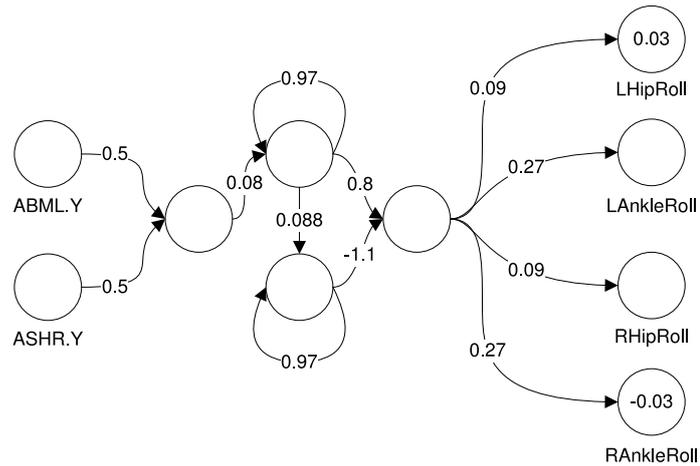


Abbildung 4.3: Reflexschleife mit Filterung der Beschleunigungswerte

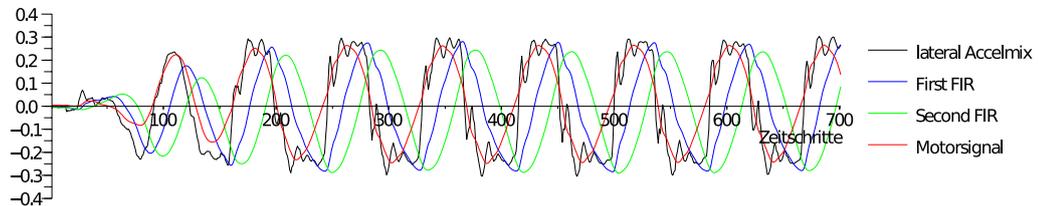


Abbildung 4.4: Output der Neuronen der erweiterten Reflexschleife während einiger Perioden der Schwingung

4.1.2 Analyse der Grundschwingung

Die Kurvenform der lateralen Beschleunigungswerte entspricht nicht der eines idealisierten, inversen Pendels. Dies wäre eine reine Sinusschwingung mit einer Frequenzkomponente. Um die Ursache der zusätzlichen Frequenzanteile zu ergründen, wird der Roboter mit einer Kamera gefilmt, die eine Framerate von 100 fps erreicht. Dies entspricht der Frequenz des SpinalCords. Die Synchronisation der Bilder mit den Daten des Roboters erfolgt nach der Aufnahme mit Hilfe der blauen LEDs, die im ersten Zeitschritt nach der Aktivierung des Netzes aktiviert wurden.

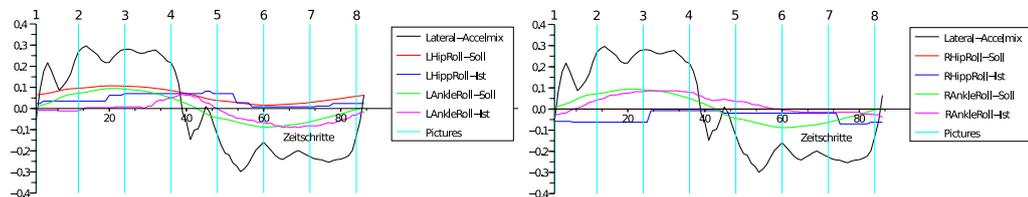


Abbildung 4.5: Eine Periode der lateralen Schwingung mit markierten Zeitpunkten der Bildaufnahmen

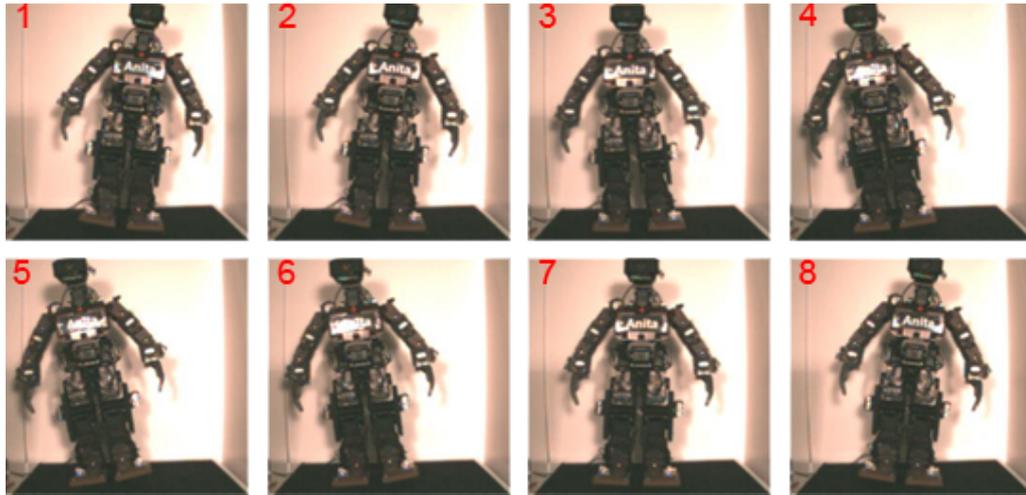


Abbildung 4.6: Auszug aus einem Hochgeschwindigkeits-Video in 120 ms Abstand. Es ist eine Periode der lateralen Schwingung zu sehen.

Der Nulldurchgang des Accelmix entspricht in etwa dem Wendepunkt der Pendelbewegung. Mit Hilfe des Videos und der Beschleunigungsdaten des Körpers lässt sich die Kurvenform genauer erklären. Auffallend ist der erste Einbruch nach einem Nulldurchgang am Zeitschritt 7 und 45. Dieser wird durch das jeweilige Bein erzeugt, welches vom Boden abhebt. Da das Bein durch das Rollgelenk in der Hüfte herangezogen wird, aber der Fuß noch auf dem Boden steht, baut sich im Bein mechanische Spannung auf, die sich entlädt, sobald der Fuß vom Boden abhebt. Der erzeugte Impuls überträgt sich auf den ganzen Körper. Die Einbrüche am Zeitschritt 20 und 60 entstehen durch das Aufsetzen der inneren Fußkante des rechten Fußes für 20 und des linken für 60. Die kleineren Einbrüche der Beschleunigung an Stelle 28 und 69 entstehen durch das komplette aufsetzen des Fußes. Sobald der Roboter über die äußere Fußkante rollt, beschleunigt er wieder.

4.1.3 Einfluss der Netzparameter auf die Schwingung

Eine andere Sicht auf den Mechanismus zur Erzeugung der Schwingung führt zur nächsten Analyse des Netzes. Das ganze System lässt sich auch als neuronaler Oszillator interpretieren, in dem ein Neuron durch den Roboter und seine Umwelt ersetzt wird. Auf verschiedenen Untergründen verändert sich die Amplitude wie auch die Frequenz des Roboters. Vor allem auf weichen Untergründen bricht die Amplitude der Schwingung ein. Verschiedene Roboter zeigen unterschiedliche Frequenz- und Amplitudenwerte, bei den gleichen Netzparametern. In diesem Abschnitt wird der Einfluss des Gewichtes der Ringkopplung r und der Selbstkopplung s des ersten Filterneurons (siehe Abbildung 4.7) auf die Periodendauer der Schwingung und der Amplitude untersucht.

Mit jedem neuen Parametersatz wird die Pendelbewegung ca. 1 Minuten lang

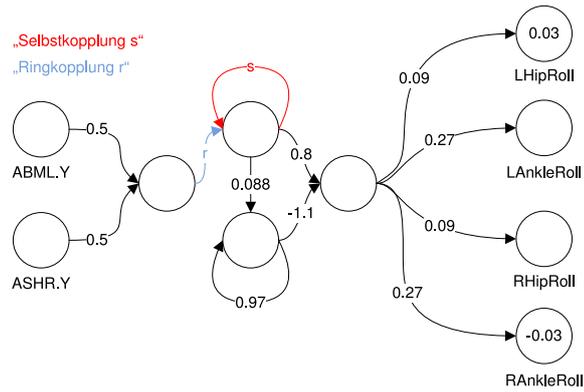


Abbildung 4.7: *Erweiterte Reflexschleife mit markierten veränderten Parametern*

gemessen. Danach wird mit Hilfe eines Scilab-Scripts die Dauer jeder Periode bestimmt. Dazu wird die laterale Schwingung mit Hilfe eines symmetrischen Filters mehrfach geglättet. Die verwendete Filtermaske sind 0.25, 0.50, 0.25. Daraufhin werden die Zeitschritte zwischen zwei Nulldurchgängen, die von negativ zu positiv gehen, gezählt. Die Amplitude wird aus dem Maximum des Betrages der entsprechenden Periode berechnet. Aus diesen Daten werden der Mittelwert und die Standardabweichung der Periodendauer und der Amplitude bestimmt.

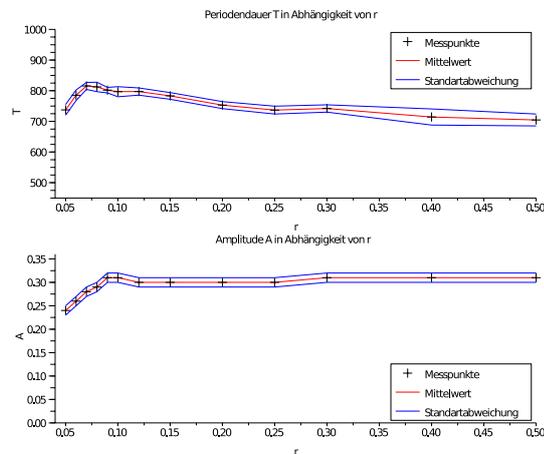


Abbildung 4.8: *Mittelwert und Standardabweichung der Periodendauer und Amplitude der lateralen Schwingung gemessen in Abhängigkeit der Verbindung r .*

In der ersten Versuchsreihe wird nur die Ringkopplung r verändert. Die Selbstkopplung s hat den Wert 0.97. Der quasiperiodische Attraktor der lateralen Schwingung stabilisiert sich erst ab einem Wert von 0.05. Die Amplitude ist mit 0.24 noch sehr gering, so dass beide Füße des Roboters durchgehend Bodenkontakt behalten. Die Periodendauer und Amplitude steigt bis 0.7 rasch an. In den Wertebereichen

0.7 und 0.8 findet sich ein kleines Plateau in dem Frequenz und Amplitude nahezu konstant sind. Mit einem Kantengewicht von 0.9 erreicht die Amplitude den maximalen Wert von 0.31, der in der ganzen Testreihe nicht mehr überschritten wird. Wird die Ringkopplung weiter erhöht, bleibt die Amplitude nahezu konstant und die Periodendauer nimmt kontinuierlich ab. Ab einem Wert von 0.2 für r tritt ein zweiter quasiperiodischer Attraktor auf, der jedoch noch instabil ist. Ab einem Wert von 0.8 stabilisiert sich dieser. Der Roboter gelangt durch schnelles Schütteln in einen höherfrequenten Attraktor. Ab einem Wert von 2.5 wechselt das System zwischen den beiden hin und her. Aus dieser Beobachtung lässt sich schließen, dass die Basins der koexistenten quasiperiodischen Attraktoren mit höherer Ringkopplung näher zusammenwachsen.

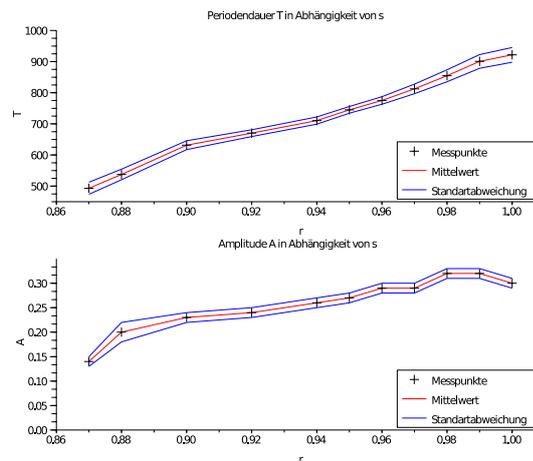


Abbildung 4.9: Mittelwert und Standardabweichung der Periodendauer und Amplitude der lateralen Schwingung gemessen in Abhängigkeit der Verbindung s .

Bei der zweiten Versuchsreihe wird die Ringkopplung auf den Wert 0.08 gesetzt und die Selbstkopplung s sukzessive verändert. Die Grenze einer stabilen Schwingung liegt hier bei einem Kantengewicht von 0.87. Die Amplitude der lateralen Beschleunigung ist noch nicht hoch genug, so dass die Füße den Bodenkontakt verlieren. Zwischen Selbstkopplung, Periodendauer und Amplitudengröße ist ein nahezu linearer Zusammenhang erkennbar.

Es ist abzusehen, dass mit Hilfe weniger Stellschrauben sich die gewünschte Frequenz und Amplitude einstellen lassen. Diese Erkenntnis kann dazu verwendet werden Lernregeln zu entwickeln, die die Schwingung stabilisieren. Somit wäre eine Feinjustierung des Netzes für die verschiedenen Roboter der A-Serie und Untergründe hinfällig. Dazu muss jedoch der Parameterraum weiter exploriert werden.

4.2 Struktur und Beschreibung des neuronalen Netzes

Bei der Beschreibung des Netzes wird in Richtung des Signalflusses vorgegangen: von den Input-Neuronen der Beschleunigungssensoren über die Hidden-Neuronen des Netzes zu den Output-Neuronen der Motoren der A-Serie.

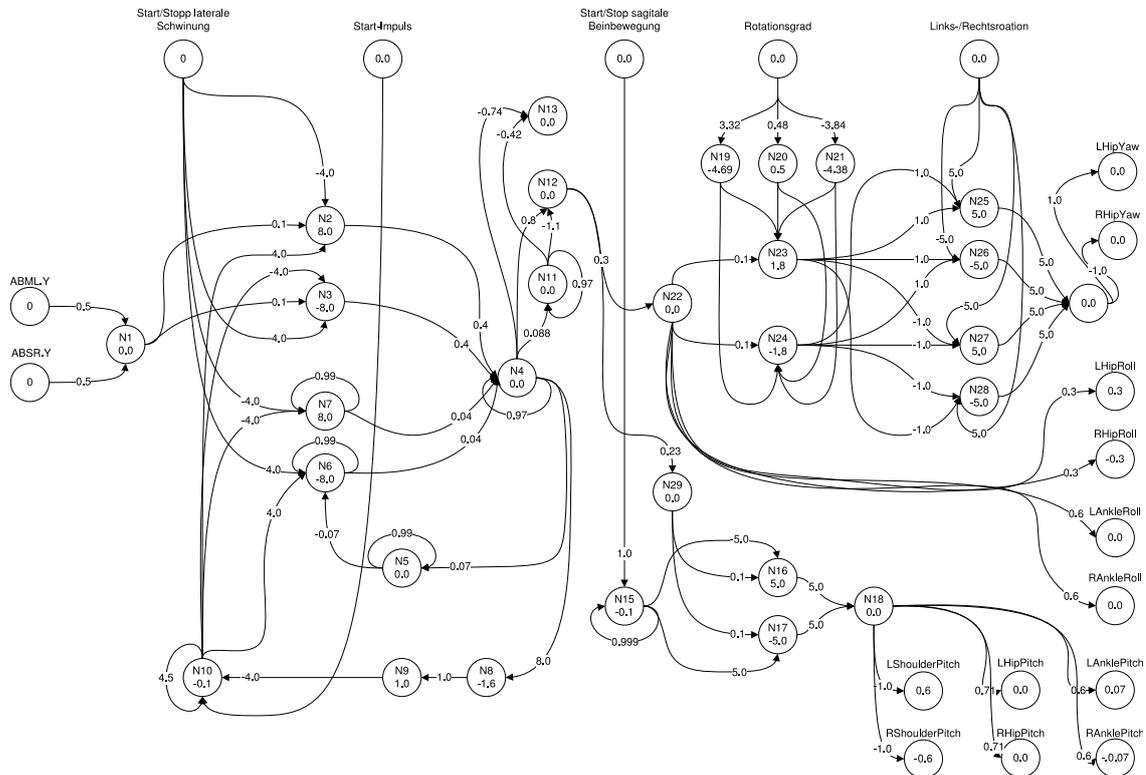


Abbildung 4.10: Vollständiges Netz zur Laufmustererzeugung der A-Serie

4.2.1 Main Driver

In diesem Abschnitt des Netzes findet die Erzeugung und Steuerung der Grundschwingung statt. N1 bildet den Mittelwert der beiden Beschleunigungswerte. N2 und N3 bilden ein Gate, welches durch das Input-Neuron „Start/Stopp laterale Schwingung“ geschlossen und geöffnet wird. Dadurch kann die laterale Schwingung jederzeit zum Erliegen und der Roboter zum Stehen gebracht werden. Es wurde ein Biaswert von acht gewählt, damit dieses Gate nur dann offen ist, wenn der interne neuronale Oszillator deaktiviert ist. Dies wird im folgenden Abschnitt näher behandelt. Der Gate-Ausgang führt in das Neuron N4, das zusammen mit dem Körper die Schwingung erzeugt, wie im Kapitel 4.1 erläutert.

4.2.2 Interner Oszillator

Das Netz befindet sich nach dem Start als erstes im Fixpunkt des Standes. Dieser Fixpunkt ist stabil. Um aus diesem in den quasiperiodischen Attraktor der lateralen Schwingung zu gelangen, benötigt der Roboter einen Impuls von außen, z.B. einen kleinen Stoß. Dies ist jedoch unbefriedigend, da der Roboter autonom interagieren soll. Daher wurde ein neuronaler Oszillator implementiert. Der Oszillator kann hinzu geschaltet werden und der Roboter fängt an zu pendeln. Eine neuronale Schaltung sorgt dafür, dass sobald sich das Netz im quasiperiodischen Attraktor befindet, der Oszillator wieder abgeschaltet wird. Den Oszillator bilden Neuron N4, N5, N6 und N7 wobei N6 und N7 gleichzeitig auch eine Gate-Funktion haben.

Die Anforderungen an den neuronalen Oszillator sind eine niedrige Frequenz von ca. 1 Hz und einer Amplitude, die groß genug ist, um den Roboter in den Einzugsbereich des quasiperiodischen Attraktors zu bringen. Dies ist mit einem $SO(2)$ -Oszillator nur schwer zu erreichen, da die Frequenz des Oszillators im Vergleich zur Updaterate sehr niedrig ist und Amplitude und Frequenz in extremen Bereichen voneinander abhängen.

N8 ist ein Schwellwertschalter der durch das Neuron N4 aktiviert wird. Der Schwellwert wurde so gewählt, dass der Schalter aktiv wird, wenn die Amplitude der Schwingung groß genug ist. Dieser Wert wurde empirisch ermittelt.

Das Neuron N9 zieht den Wert von N8 in den Bereich $]0; 1[$.

N10 besitzt eine Rückkopplung von 3.5 und ist somit ein Hysterese-Neuron. Der Biaswert von 0.1 bewirkt, dass nach dem Start des Netzes das Neuron in den Fixpunkt -1 wandert. Dadurch ist der Oszillator zuerst durch das Gate N6 und N7 deaktiviert. N10 kann mit der Hilfe des Input-Neurons „Start-Impuls“ in den positiven Fixpunkt gebracht werden, der Oszillator ist somit aktiviert und die Beschleunigungswerte werden abgekoppelt. So kommt es zu einer kontrollierten Schwingung des Oszillators und damit auch des Roboters. Wird der Schwellwert erreicht, zieht N9 N10 wieder in den negativen Bereich, der Oszillator wird deaktiviert und die laterale Beschleunigung wieder eingekoppelt. Das System befindet sich nun im stabilen quasiperiodischen Attraktor. Die Biaswerte der Gate-Neuronen betragen acht, da immer zwei Bedingungen erfüllt sein müssen, damit sich das jeweilige Gate öffnet. Diese sind für das Oszillator Gate (N6 und N7) zum einen ein Signalwert von Eins des Input-Neurons „Start/Stopp laterale Schwingung“ und zum anderen muss N10 sich im Positiven Fixpunkt nahe der Eins befinden. Das Gate (N2 und N3) öffnet sich wenn „Start/Stopp laterale Schwingung“ eine Eins liefert und am Ausgang von N10 minus Eins anliegt.

In Abbildung 4.11 sieht man wie der Oszillator aktiviert wird. Nachdem das System ein ausreichend große Schwingungsamplitude erreicht hat, wird der Oszillator wieder abgeschaltet. Ab diesen Zeitpunkt schwingt das System von selbst weiter.

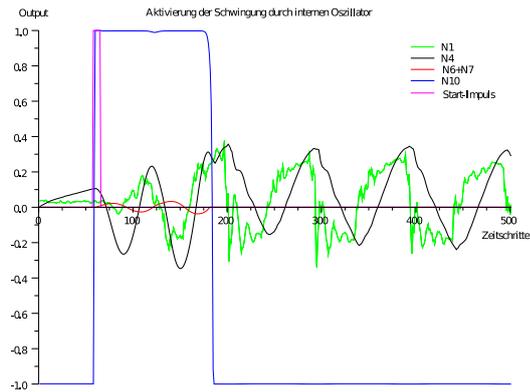


Abbildung 4.11: Output der lateralen Beschleunigung des Motorneurons und des internen Oszillators nach „Start/Stopp“-Impuls

4.2.3 Mischpult

In diesem Abschnitt des Netzwerkes werden aus N4 und N11 in N12 und N13 zwei phasenverschobene neue Schwingungen generiert. Die theoretische Grundlage bildet die Tatsache, dass zwei sinusförmige Schwingungen addiert eine in der Amplitude und Phase veränderte Schwingung ergeben.

$$x(t) = A \cdot \sin(\omega t + \phi_1) \quad A: \text{Amplitude}, \phi: \text{Phase} \quad (4.1)$$

$$= \alpha \cdot \sin(\omega t + \phi_2) + \beta \cdot \sin(\omega t + \phi_3) \quad (4.2)$$

Die Gewichte wurden so gewählt, dass N12 in der Phase und Amplitude mit der Grundschiwingung der Beschleunigungswerte übereinstimmt. Der Output von N12 ist somit nahezu von allen höherfrequenten Komponenten der Schwingung der Beschleunigungswerte bereinigt. N12 hat den selben Output wie N13, allerdings um 90° phasenverschoben. Dies erleichtert ein späteres Mischen von phasenverschobenen Schwingungen, da die Additionstheoreme von Sinus und Cosinus angewandt werden können.

4.2.4 Einkopplung der Vorwärtsbewegung

N16 und N17 bilden ein Gate, das die Vorwärtsbewegung des Roboters steuert. Das Gate wird über das Inputneuron „Start/Stopp sagittale Beinbewegung“ geöffnet und geschlossen. N15 ist ein starker Tiefpassfilter, der verhindern soll, dass die Vorwärtsbewegung zu ruckartig eingekoppelt wird und der Roboter dabei fällt. Dies ist besonders gefährlich, wenn zum Zeitpunkt der Gate-Öffnung das Ausgangssignal von N14 den Wert der maximalen Schwingungsamplitude besitzt. Über N18 wird das Signal an die Pitch-Gelenke der Beine und Arme des Roboters verteilt. Die Gewichte der Verbindungen zu den einzelnen Motor-Neuronen wurden per „Trial and Error“ ermittelt.

4.2.5 Steuerung der Links-Rechts-Rotation

N19 bis N24 bilden den linearen Multiplikator der über das Inputneuron „Rotationsgrad“ gesteuert wird. Durch die Amplitude des Signals, welches den linearen Multiplikator verlässt, wird der Radius der Kurve gesteuert. N25, N26 und N27,N28 bilden jeweils ein Gate. Sie werden durch das Input-Neuron „Links-Rechts-Rotation“ exklusiv geöffnet und geschlossen. Der Output dieser Gates wird in dem Neuron N29 zusammengefasst. Die Gewichte der Verbindungen von N27 und N28 zu N29 sind negativ. Dies bewirkt eine Phasenumkehr der Schwingung. Je nachdem welches Gate nun offen und welches geschlossen ist kommt es zu einer Links- oder Rechtsdrehung. Das Signal von N29 wird an die Yaw-Hüftgelenke verteilt. Dabei wird es zur rechten Hüfte in der Phase gedreht. So kommt eine Drehung des Roboters während des Gehens zustande.

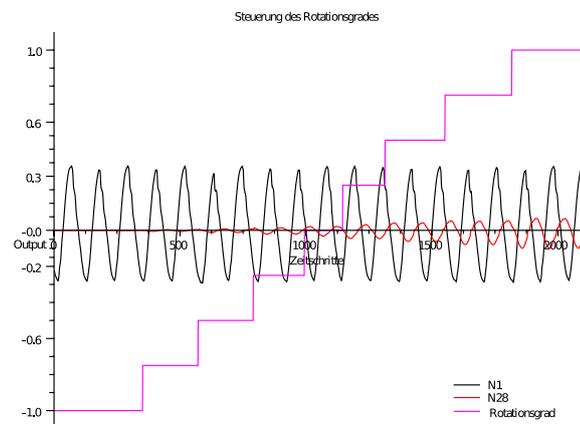


Abbildung 4.12: Steuerung des Rotationsgrades während des Gehens

In Abbildung 4.12 ist der Einfluss des Input-Neurons „Rotationsgrad“ auf die Amplitude des Signals zu erkennen. Der lineare Zusammenhang wird sehr gut sichtbar.

Zusammenfassung und Ausblick

In dieser Studienarbeit wurde zuerst auf die Keyframetechnik eingegangen. Anschließend folgte ein Abriss über den neuronalen Central Pattern Generator, wobei seine adaptiven Fähigkeiten mithilfe von Reflexschleifen herausgearbeitet wurden. Als Letztes wurde gezeigt, dass es mit Hilfe einer einfachen sensomotorischen Schleife und den vorgestellten Neuromodulen möglich ist ein Gangmuster zu erzeugen. Der so entwickelte neuronale Controller ermöglicht dem Roboter von selbst loszugehen und wieder stehenzubleiben. Weiterhin können Links- und Rechtskurven mit unterschiedlich starken Radien gegangen werden. Durch die Ein- und Entkopplung der sagittalen Beinbewegung ist der Roboter in der Lage sich auf der Stelle zu drehen. Das Netz ließe sich sehr leicht um eine Schrittweitensteuerung erweitern. Dazu müsste nur das Gate welches die sagittale Beinbewegung an und abschaltet durch einen linearen Multiplikator ersetzt werden. Auch wäre durch eine Phasenverschiebung des Signals der sagittalen Beinbewegung und einer Gewichtsverlagerung ein Rückwärtsgehen realisierbar. Erste Versuche dazu wurden bereits unternommen. Wie in Kapitel 4.1 gezeigt wurde lässt sich durch die Veränderung zweier Gewichte sowohl die Amplitude als auch die Frequenz der lateralen Schwingung einstellen. Mit dieser Voraussetzung ließe sich eine homöostatische Lernregel entwickeln. So könnte manuelle Feinjustierung des neuronalen Netzes für verschiedene Roboter der A-Serie und für unterschiedlichen Bodenbelag obsolet gemacht werden. Bei der sagittalen Beinbewegung ist noch viel Handlungsbedarf. Zwar führt die einfache Einkopplung der lateral erzeugten Schwingung zum gewünschten Ergebnis, doch ist die Schrittgeschwindigkeit des Roboters noch immer langsam. Mit 7,5 cm/s ist es um die Hälfte schneller als das Gangmuster welches mit Hilfe der Keyframe-Technik erzeugt wird. Aber es ist noch weit von der Geschwindigkeit eines Menschen entfernt, der auf 47 cm skaliert in etwa auf 32 cm/s kommen würde. Auch wird der Roboter sobald er beginnt vorwärts zu gehen instabiler. Durch die „blinde“ Beinbewegung der Pitch-Gelenke kommt es häufig vor, dass der Roboter über seine eigenen Füße stolpert. Denkbar wäre eine Erweiterung um eine extra Reflexschleife für die sagittale Bewegung der Beine. Eine Implementierung der stabilisierenden Reflexe wie sie in [Pra00] vorgestellt werden, könnte zu einer Stabilisierung des Roboters führen. Somit ist das Ziel für die nähere Zukunft weitere Reflexschleifen zur Stabilisierung und Erhöhung der Geschwindigkeit in das bestehende Netz zu integrieren.

Literaturverzeichnis

- [AC99] Davide Boothe Avis Cohen. Sensorimotor interactions during locomotion: Principles derived from biological systems. *Autonomous Robots*, 1999.
- [AS85] Maurice Moulins Allen Selverston. Oscillatory neural networks. Technical report, 1985.
- [EM01] Dirk Bucher Eve Marder. Central pattern generators and the control of rhythmic movements. *Current biology*, 2001.
- [FP03] Keyan Zahedi Frank Pasemann, Manfred Hild. So(2)-networks as neural oscillators. In *Computational Methods in Neural Modeling*. Springer Berlin / Heidelberg, 2003.
- [GE04] J. Nakanishi G. Cheng G. Endo, J. Morimoto. An empirical exploration of a neural oscillator for biped locomotion control. *Robotics and Automation*, 2004.
- [GE08] Takamitsu Matsubara Gen Endo, Jun Morimoto. Learning cpg-based biped locomotion with a policy gradient method: Application to a humanoid robot. *The International Journal of Robotics Research*, 2008.
- [Har79] Daniel Hartline. Integrative neurophysiology of the lobster cardiac ganglion. *American Zoologist*, 1979.
- [Hei07] Daniel Hein. Evolution of biped walking using physical simulation., 2007.
- [Hil07] Manfred Hild. *Neurodynamische Module zur Bewegungssteuerung autonomer mobiler Roboter*. PhD thesis, Humboldt-Universität zu Berlin, 2007.
- [HK07] Avis H. Cohen Hiroshi Kimura, Yasuhiro Fukuoka. Biologically inspired adaptive walking of a quadruped robot. *Philosophical Transactions*, 2007.
- [Hoo00] Scoot Hooper. Central pattern generators. Technical report, 2000.
- [Kat73] I. Kato. Hydraulically powered biped walking machine with a high carrying capacity. *Proceedings of the 4th International Symposium on External Control of Human Extremities.*, 1973.
- [KH] Y. Haikawa T. Takenaka K. Hirai, M. Hirose. The development of humanoid robot.

- [KK02] Shuuji Kajita Kazuhiko Yokoyama Kenji Kaneko, Fumio Kanehiro. Design of prototype humanoid robotics platform for hrp. In *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, 2002.
- [Kuo02] Arthur Kuo. The relative roles of feedforward und feedback in the control of rhythmic movements. *Motor Control*, 2002.
- [Lew02] Anthony Lewis. Gait adaption in a quadruped robot. *Autonomous Robots*, 2002.
- [Mat85] Kiyotoshi Matsuoka. Sustained oscillations gernated by mutually inhibiting neurons with adaption. Technical report, 1985.
- [Pra00] Jerry E. Pratt. *Exploiting Inherent robustness and Natural Dynamics in the Control of Biedal walking Robots*. PhD thesis, Massachusetts Institute of Technology, 2000.
- [Rei85] R. Reiss. A theory and simulation of rhythmic behavior due to reciprocal inhibition in nerve nets.s, 1985.
- [Thi07] Christian Thiele. Integrierte entwicklungs Umgebung zur bewegungssteuerung humanoider roboter., 2007.
- [Vuk04] Miomir Vukoratic. Zero moment point – thirty five years of its life. *International Journal of Humanoid Robotics*, 2004.
- [YK03] T. Ishida K. Nagasaka J. Yamaguchi Y. Kuroki, M. Fujita. A small biped entertainment robot exploring attractive applications. *Robotics and Automation*, 2003.