Doctoral Thesis

Walking gait control for four-legged robots
bio-inspired technology analyzed and applied to robotics

Author(s):
Still, Susanne

Publication Date:
2000

Permanent Link:
https://doi.org/10.3929/ethz-a-004036719

Rights / License:
In Copyright - Non-Commercial Use Permitted
WALKING GAIT CONTROL FOR FOUR-LEGGED ROBOTS
Bio-inspired technology analyzed and applied to Robotics

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH
for the degree of

Doctor of Natural Sciences

presented by

Susanne Still
Dipl. Phys., University of Hannover
born on the 26th of July 1970
citizen of Germany

accepted on the recommendation of
Prof. Dr. Rodney J. Douglas, examiner
Prof. Dr. Klaus Hepp, co-examiner

21.6.2000
Dedicated to the memory of

Misha Mahowald
Abstract

The objective of this thesis is the design and analysis of a novel, compact controller for the inter-leg coordination of four-legged walking machines. The controller is implemented using Very Large Scale Integrated (VLSI) technology. It is able to generate walking behaviors for a four-legged machine, including all of the walking gaits frequently observed in four-legged animals, and to control turning maneuvers, without the need of any software. The controller can be interfaced to both sensory devices and higher order controllers, digital or analog. It is demonstrated that this controller is useful for applications on a real robot. Further contributions of this thesis are (i) a systematic analysis of a previously invented class of discrete-component walking controllers and two robotic applications using controllers of this type, and (ii) the demonstration of walking gait learning in a novel way using the chip interfaced to a digital machine that is running an unsupervised Support Vector learning algorithm.

Inter-leg coordination means the control of the phase relationships between the cyclic movements of individual legs during a rhythmic motor behavior that leads an animal or a robot to successful locomotion. The phase relationships between the legs determine the walking gait that is used. Four-legged animals display a variety of distinct walking gaits. Different walking gaits are used in different situations, distinguished, for example, by the animals speed or the terrain that is being traversed. The inter-leg coordination of a walking machine is necessary for the functioning of the machine. A controller that is furthermore able to coordinate the legs in more than one way enables the robot to use different gaits in different situations, as animals do.

The first part of the thesis (Chapter 2) is devoted to the mathematical analysis of biologically inspired networks of oscillators implemented with discrete components that have been used previously, in a rather empirical manner, to control walking machines (but not with many different gaits). The behavior of these oscillator networks is clarified. Specifically, the pattern generating abilities of these circuits are analyzed with respect to applications for walking machines, and the minimal architecture is found for a network that is able to control the inter-leg coordination of a four-legged robot such that all the standard animal walking gaits can be generated. In the remainder of Chapter 2, two different oscillator-networks of this class are implemented on robots and the resulting behaviors are recorded.

In the second part of the thesis (Chapter 3), the Walking Gait Control (WGC) chip is introduced. The chip is neuromorphic in the sense that it mimics functions of the neuronal control circuitry underlying the locomotion of animals. The circuits on the chip are analyzed mathematically and general results are found concerning the pattern generating abilities of ring versus chain control architectures. Specif-
ically, the same minimal network architecture as in Chapter 2 achieves inter-leg coordination such that all the standard animal walking gaits can be controlled. The behavior of the chip depends on a set of analog bias voltages. These are the inputs to the chip and are also referred to as the control parameters of the chip. Changing the input voltages causes the chip to generate another movement pattern. This means that the chip reduces the problem of coordinated locomotion control to a less complex problem, namely the setting of control parameters. The control parameters can be set by any device that is capable of generating analog output voltages. Such devices include other neuromorphic chips that perform sensory (pre-)processing. Digital computers can also be interfaced to the WGC chip, using commercially available hardware.

The chip's performance on a four-legged robot is tested in Chapter 4. A walking machine is constructed for this purpose and a simplified model of the machine's walking behavior is given. The chip successfully controls all of the standard quadrupedal walking gaits on this robot.

The last part of the thesis (Chapter 5) deals with learning of movement patterns. The WGC chip that controls the walking robot is interfaced to a digital computer. The resulting robotic system is able to learn desired movement patterns, i.e. walking gaits, without the intervention of an experimenter. The use of the WGC chip reduces the motor learning task to a classification problem in a natural way: find those regions in the chip's input parameter space that lead to a desired gait. The machine is given a specification of the desired gait. An algorithmic procedure is introduced that creates unlabeled examples and uses them as training data to solve the classification problem. The method is based on a recently introduced unsupervised Support Vector learning algorithm.
Kurzfassung


Gangartenkontrolle wird durch die Koordination der einzelnen Beinbewegungen gewährleistet. Verschiedene Gangarten unterscheiden sich durch die Grösse der relativen Phasenverschiebungen zwischen den zyklischen Bewegungen der Beine voneinander. Vierfüsser zeigen eine Reihe unterschiedlicher Gangarten, die sie in verschiedenen Situationen, je nach Geschwindigkeit und überquertem Terrain, nutzen. Damit eine Laufmaschine sich überhaupt vorwärts bewegt, müssen die Bewegungen der Beine auf mindestens eine Weise relativ zueinander sinnvoll koordiniert werden. Eine Steuerung, die dem Roboter darüber hinaus mehrere verschiedene Gangarten ermöglicht, ist vorteilhaft, da sie es dem Roboter ermöglicht je nach Situation unterschiedliche Gangarten zu benutzen, so wie Tiere das ebenfalls tun.

Im ersten Teil der Arbeit (Kapitel 2) werden spezielle, biologisch inspirierte Schaltkreise analysiert, die aus Netzwerken von gekoppelten Oszillatoren bestehen, welche mit diskreten Komponenten realisiert wurden und schon zuvor zur Kontrolle von Laufmaschinen (allerdings nicht zur Gangarten-Kontrolle) eingesetzt wurden. Insbesondere wird analysiert, welche Architekturen dieser Oszillator-Netzwerke ein für die Gangartenkontrolle von Laufmaschinen nützliches Verhalten aufweisen. So wird innerhalb dieses Ansatzes die minimale Kontroll-Architektur gefunden, die dazu in der Lage ist, alle standardmäßig beobachteten Gangarten von Vierfüßlern zu modellieren.


\(^1\)Dieses Wort ist im deutschen Sprachgebrauch nicht weit verbreitet. Es werden damit Systeme bezeichnet, die neuronale Rechenprozesse nachahmen und die in den meisten Fällen in aVLSI chips implementiert sind.
Blank Leer / Seite Leer /
... take a walk on the wild side ...

- Lou Reed
Contents

1 Introduction
   1.1 Quadrupedal Walking Gait Control ........................................ 2
   1.1.1 Experimental Observations ............................................. 4
   1.1.2 Theoretical Approaches ................................................. 6
   1.2 Previous work ................................................................. 9
   1.2.1 Computer-controlled four-legged walking robots .................... 9
   1.2.2 Neuromorphic chips for motor control .................................. 10

2 Discrete component controllers .................................................. 14
   2.1 Introduction ........................................................................ 15
   2.2 Analysis of the circuits ...................................................... 16
      2.2.1 Notation .................................................................. 16
      2.2.2 The Oscillator .......................................................... 17
      2.2.3 The Coupling ............................................................. 21
   2.3 Patterns produced by oscillator networks .................................. 27
      2.3.1 Rings of oscillators ....................................................... 28
      2.3.2 Chains of oscillators ..................................................... 31
      2.3.3 Rings of coupled threshold elements ................................ 33
   2.4 Sensory modulatory input ..................................................... 40
   2.5 Applications ...................................................................... 42
      2.5.1 The Robot and its Sensors .............................................. 42
      2.5.2 Data Acquisition .......................................................... 42
      2.5.3 Walking controller made from a chain of two coupled oscillators 44
      2.5.4 Locomotion of the Walking Machine ................................ 45
      2.5.5 Walking controller made from a ring of four threshold elements 46
      2.5.6 Locomotion of the Walking Machine ................................ 47
      2.5.7 Discussion .................................................................. 51
   2.6 Conclusion ........................................................................ 52

3 Walking Gait Control Chip .............................................................. 53
   3.1 The Chip .......................................................................... 53
   3.2 Data Acquisition ............................................................... 53
   3.3 Circuit Analysis .................................................................. 57
      3.3.1 Notation .................................................................. 59
3.3.2 The Oscillator ............................................. 60
3.3.3 The Coupling ............................................. 62
3.3.4 Temperature dependency ................................. 67
3.4 Patterns produced by rings and chains of oscillators .... 67
  3.4.1 Chains of oscillators .................................... 67
  3.4.2 Rings of oscillators ..................................... 76
3.5 Sensory feedback ............................................ 81
3.6 Conclusion .................................................... 84

4 Four-Legged Walking Machine ................................. 86
  4.1 The Robot .................................................. 86
  4.2 Gait Control using the WGC chip ......................... 88
  4.3 Kinematic Model of the Robot ............................. 98
     4.3.1 Forward motion ........................................ 101
     4.3.2 Walking on a curved trajectory ...................... 102
  4.4 Summary .................................................... 108

5 Learning Walking Gaits .......................................... 109
  5.1 Learning procedure ......................................... 111
     5.1.1 Creating the training data for the learning algorithm 112
     5.1.2 Learning algorithm ................................... 112
  5.2 Implementation ............................................. 117
  5.3 Learning different movement sequences .................... 118
     5.3.1 The Walk Gait .......................................... 119
     5.3.2 The Transverse Gallop ................................. 123
     5.3.3 Walking on a curved trajectory ...................... 123
  5.4 Summary .................................................... 123

6 Conclusion and Outlook ........................................ 127
  6.1 Achievements ............................................... 127
  6.2 Problems and possible solutions ........................... 129
  6.3 Possible extensions ....................................... 129
     6.3.1 Extending the WGC chip’s control capabilities ....... 129
     6.3.2 Extending the system’s learning capabilities ........ 130
     6.3.3 Extending the system’s sensory input ................ 131

Bibliography ...................................................... 133
Chapter 1

Introduction

Locomotion, meaning the progressive motion of an animal, is often the most complex motor behavior performed in a given species [43]. It is a non-trivial task, crucially important for the animal’s survival, involving the coordination of a large number of muscles [43]. Locomotion is a highly adaptive behavior [125]. Through sensory feedback, loops are created which involve the environment and the moving creature. Locomotion and the control mechanisms underlying locomotion are often viewed as a model system for understanding the complex functioning of the nervous system [65]. Besides walking, locomotion also includes swimming, flying, crawling and snake-like propulsion. This thesis, however, is concerned only with four-legged locomotion.

Legged animals use distinct walking gaits [3] [84]. Gait transitions help an animal to lower its energy consumption at a given speed [58]. Similarly, a legged robot can make use of different walking gaits depending on its speed and on the terrain. The physical shape of a walking robot might make some walking gaits more appropriate for a given terrain and at a given speed than others, not only for reasons of energy consumption, but also because a particular task, e.g. turning, might be simplified.

Legged land animals can reach considerable speeds (the fastest being the cheetah, *Acinonyx jubatus*, at ca. 70 mph [98]) or can move very slowly as tortoises do [61]. Many legged animals (e.g. squirrels, monkeys) can also use their front paws for fine manipulations. Currently available walking machines lag far behind the flexibility and efficiency of walking animals. Handling rough terrain without causing major damage to the terrain is very difficult for wheeled or tracked vehicles. Legged robots can be useful in situations where it is impractical or not desirable to flatten the terrain to create a path for the machine. Legged machines are used, for example, for forest work and for de-mining mine-fields [33]. They could also potentially be used as supplements for wheel chairs [14] or as prostheses. The ultimate goal is to build walking robots which can perform locomotion tasks in an intelligent and autonomous way as legged animals do.
In this context, it is desirable to control the movement of legged robots in a way that is similar in essence to the way that walking behavior is controlled in animals. The control of robotic locomotion places a problem, because the coordination required for locomotion is complex. Algorithmic approaches face difficult problems when they have to deal with many sensors and many actuators [32]. It is necessary to develop an approach which scales with the number of actuators and the number of sensors without becoming too complex and therefore too slow or too difficult to implement. This thesis describes such an approach, implementing principles of biological locomotion control in a low cost, low power and low weight system (see Chapter 3). As Lewis et al. [72] recently pointed out, challenges for the future of walking machines include miniaturization and performance in real-time. This requires small, low-cost and power-efficient controllers [72].

The problem of controlling legged machines has been a subject of research for several decades (see Sec. 1.2.1). In the present work, a Walking Gait Control (WGC) chip is introduced that is inspired by biological findings (see Chapter 3). The chip is based on results obtained from the analysis of a discrete implementation of similar controllers (see Chapter 2). The WGC chip drives a four-legged robot without any interface to a digital computer and can incorporate sensory feedback (see Sec. 3.5). However, the WGC chip can also be interfaced to a digital computer resulting in a flexible, hybrid system. To demonstrate the utility of this robotic system, an unsupervised learning algorithm (see Chapter 5) running on the digital computer is used to automatically fine-tune the parameters of the WGC chip such that the robot learns to walk with a desired walking gait.

In the following, the work described in this thesis is motivated in the context of some background information on experimental findings and theoretical models concerning walking gait control in animals (see Sec. 1.1) and in the context of previous related work in robotics and neuromorphic engineering (see Sec. 1.2).

1.1 Quadrupedal Walking Gait Control

Different gaits are distinguished for quadrupedal locomotion, according to the patterns of inter-limb coordination. Muybridge [84] classified eight different gaits based on the observation of 26 different wild and domesticated four-legged animals. These gaits are the walk, the amble, the trot, the pace, the canter, the transverse and the rotary gallop, and the ricochet. The ricochet is a gait used by kangaroos, involving only two of the four legs of the animal. It was not included in the classification of four-legged gaits by Alexander [2]. He distinguished gaits by the phase relationships between the legs (see Fig. 1.1). Since the amble is a fast walk and has the same phase relationships as the walk, he did not list it as a distinct gait. However, he included the bound gait and the pronk gait. This distinction between quadrupedal
gaits based on the phase relationships between the animal’s feet was adopted later by other researchers (see for example [109] and [104]) who tried to model gaits and gait transitions mathematically (see Sec. 1.1.2). In the present thesis, this classification likewise serves as a definition for different quadrupedal gaits.

The walk gait is used by many animals (e.g. cats, horses, dogs, camels, elephants and cloven-footed animals like the ox [84]) for slow locomotion. The animal’s feet fall in the succession left front (LF), right hind (RH), right front (RF) and left hind (LH) with phase lags of 0.25 (where 1 denotes a full cycle) between each of them.

Animals solve an optimization problem when they move. On one hand, minimizing the energy cost for locomotion is advantageous for the animal, on the other hand, the movement sequence has to guarantee that the animal does not fall. The slower the gait, the more important it becomes to maintain a stable equilibrium throughout the whole movement sequence. One might expect that tortoises, which are very slow and therefore need to maintain equilibrium, would use a gait that ensures stability, such as the walk gait. In fact they move using a gait with the same succession.
of footfalls (LF, RH, RF, LH) but with phase lags of 0.1 between LF and RH, 0.4 between RH and RF and 0.1 between RF and LH [2]. Doing this they rise and fall, pitch and roll during the movement [2]. The reason for choosing this gait seems to be that their muscles are slow and therefore cannot exert the forces needed to maintain a more stable gait ([61] and [1]). Slow muscles can maintain tension at a lower energy cost than fast muscles [2]. This is an advantage for the tortoise which neither has to chase prey nor to run from predators. The gait that the tortoise uses is optimal with respect to stability given the slow characteristics of the tortoise’s muscles [2].

The trot gait is also used by many animals (e.g. cats, horses, dogs, deer, oxen and their respective seriates [84]). Diagonally opposite feet are in phase with each other and there is a phase lag of 0.5 between the pairs LF/RH and RF/LH. Elephants typically use the amble instead of the trot and do not have any faster gaits [2]. Camels use the pace gait. Here, the legs on each side of the animal are in phase while there is a phase lag of 0.5 between the pairs LF/LH and RF/RH.

A horse, as it increases its speed, changes from a walk to a trot to a canter and finally to a gallop to lower its energy consumption [58]. In the canter, LF and RH move in phase, RF moves with a phase lag of 0.3 relative to LF and finally LH moves with a phase lag of 0.4 relative to RF. Cats, horses, goats, camels, dogs and many other animals [84] use the gallop for fast locomotion. Two different types of gallop are distinguished, the transverse gallop and the rotary gallop. In both gaits, there is a 0.1 phase shift between the two front legs and also between the two hind legs. There is a 0.5 phase shift between LF and one of the hind legs, namely the LH (transverse gallop) or the RH (rotary gallop).

In the bound gait, which is used for example by the Siberian souslik [35], both front legs move in phase and both hind legs move in phase, with a phase lag of 0.5 between the legs of front and hind girdle. In the pronk gait, all legs move in phase.

In the remainder of this section, experimental evidence is discussed (see Sec. 1.1.1) which indicates how the inter-limb control that leads to different walking gaits can be modeled theoretically (see Sec. 1.1.2).

1.1.1 Experimental Observations

In the first half of the nineteenth century, two hypotheses were put forward to account for the generation of rhythmic movement patterns (particularly those used to control locomotion) by the nervous system [41]. One hypothesis proposed that rhythmic motor behavior is generated by centers in the nervous system, which are referred to as Central Pattern Generators (CPGs) or central rhythm generators. The other hypothesis proposed that a chain of reflexes, elicited by peripheral feedback, would be responsible for generating rhythmic motion. For decades, a controversial
debate took place between adherents of these two views. Experimental evidence in favor of the CPG hypothesis was found in deafferentiation, paralysis and isolation experiments. The latter method clearly provides the most convincing evidence. All or part of the nervous system is isolated from the rest of the animal's body. If the firing pattern recorded in the stumps of appropriate nerve bundles remains coordinated this must mean that the nervous system is capable of generating phased output patterns without being attached to sensors, which would clearly support the CPG hypothesis. A remarkable body of evidence for the CPG hypothesis was collected for many animals of different species, vertebrates and invertebrates, and for a variety of movements, amongst them walking, swimming and breathing (for a review see [29]).

The main argument against the CPG hypothesis was that in individual cases reflexes can produce rhythmic movement patterns and deafferentiation causes such motor patterns to cease. In the 1930's and 1940's, findings of Gray and collaborators showed that leeches, for example, responded during swimming in such a way that each movement seemed to trigger the next movement such that the entire behavior could be viewed as a chain of reflexes [40]. Furthermore, they did not find rhythmic motor output in the isolated nerve cords of these animals. The latter observation was falsified in the 1970's by Kristan and Calabrese [69]. Similarly, reports by Gray and Lismann stating that deafferentiated spinal cords of toads did not produce coordinated output were falsified in 1970 by Harcombe and Wyman [49]. Together with the successful isolation experiments in a variety of other animals, this suggests that Gray et al. would have found rhythmic motor patterns in the absence of sensory input, had they treated the preparations carefully enough [29].

In the early 1980's, Delcomyn [29] concluded in his review of the previous controversy that the experimental evidence showing that there are regions in the nervous system that do generate rhythmic patterns in the absence of sensory input is overwhelming. However, he acknowledges the fact that sensory feedback can stabilize and in some cases considerably alter motor behavior. In the last few decades, considerable progress has been made on the identification of the neurons that constitute those circuits which generate rhythmic output for locomotion (see e.g. [64]). It should be mentioned that even today models exist which intend to show that reflexes triggered by sensory feedback are sufficient to model walking behavior [27].

Grillner et al. revealed the circuitry which coordinates the swimming behavior of the lamprey and provided a detailed cellular analysis of the neural network (see e.g. [47], [44] and [42]). Distinct excitatory and inhibitory populations of interneurons were identified. These, together with motoneurons can be modeled as neuronal oscillators. One oscillator controls the muscles of one body segment. The oscillators are coupled yielding relative phasing of the movements of the segments, which in turn leads to the trunk movement that the lamprey uses to swim. Stretch receptors
influence the neuronal oscillators in preparations with intact afferents. However, these are not necessary for the production of a rhythmic, neuronal motor output [45].

Walking involves the coordination of multi-jointed movements of multiple limbs. In higher vertebrates, it is more difficult to experimentally determine the neuronal circuitry underlying locomotion. Working from indirect evidence, Grillner attempted to sub-divide CPGs into so called unit burst generators [41] proposing that there should be one such generator for each limb. Reciprocal interactions between different limb centers [82] would then control the phasing between the legs, i.e. the gait. With currently available experimental techniques, this hypothesis seems to be difficult to test, particularly for mammals, due to the complexity of the mammalian spinal cord [19].

Recent work on an amphibian, the mudpuppy (Necturus maculatus), has revealed that the neural network for walking which controls a single joint (the elbow) of a limb contains at least two separate rhythm generators, a flexor center, and an extensor center which excite flexor and extensor motor neurons, respectively [19]. The two centers can work independently. Mutual inhibitory connections between them are important for the coordination between flexor and extensor. Both centers receive excitatory and inhibitory sensory inputs which can reset and profoundly modulate the centrally generated rhythm.

Adaptive changes in inter-limb coordination were shown not to take place in the spinal cord [125]. Rather they are assumed to happen in the brainstem and the cerebellum.

1.1.2 Theoretical Approaches

Models of walking gaits based on coupled oscillators can be distinguished by the way in which gait transitions are controlled [21]. In one class of models, transitions are induced by changing one global parameter that affects the whole network simultaneously, in the other class, the relative coupling strengths between individual oscillators are changed locally.

The most general approach to gait control, which is independent of the details of the oscillator dynamics and the coupling between oscillators, is the group-theoretic approach. This approach examines how the symmetry of networks of coupled nonlinear oscillators leads to oscillatory patterns. The onset of an oscillatory pattern is modeled as a symmetric Hopf bifurcation [38], while transitions between different patterns are modeled as various symmetry-breaking bifurcations. The bifurcation parameter can be chosen to be the coupling strength. Collins and Stewart showed [23], using this approach, that patterns which can be obtained via Hopf bifurcation in networks of four symmetrically coupled nonlinear oscillators correspond to many
of the observed quadrupedal gaits, assuming that the signal from one oscillator is transferred to one leg. An extension of this work uses networks of $4n$ oscillators to model $2n$-legged locomotion [36] to overcome difficulties encountered with the networks of four symmetrically coupled nonlinear oscillators. One shortcoming of the four-oscillator networks is that it is necessary to use several different architectures in order to obtain all of the standard quadrupedal walking gaits (which are shown in Fig. 1.1). Golubitsky et al. [37] (for further details see [16] and [15]) showed that a network of eight symmetrically coupled nonlinear oscillators can generate patterns that correspond to the standard quadrupedal walking gaits.

Another important weakness of the four-oscillator networks is that trot and pace occur as symmetrically related periodic solutions, also called *conjugate* periodic solutions [36]. Conjugate periodic solutions exist simultaneously and should have the same stability properties. If two different gaits are modeled by conjugate periodic solutions, the gaits should also coexist in the same animal under similar conditions and have the same stability properties. This is contradicted by experimental evidence ([9], [39] and [55]). The eight-oscillator network can model trot and pace without these unwanted conjugacies [36].

The work of Schöner et al. [104] which uses a synergetic approach, similarly describes symmetries of walking gaits group-theoretically. Schöner et al. obtained gait transitions as phase transitions in the dynamical systems they analyzed. The group-theoretic approach to CPGs is interesting because it reveals general connections between the symmetry of the gaits and the symmetry of the underlying networks. The assumption that a global control parameter is changed to induce gait transitions is partly supported by some physiological evidence that the output of a motor CPG can be modified by changes to its descending inputs [105] and by changes to afferent inputs from peripheral sensory organs [89].

Another physiologically plausible approach was formulated first by Grillner [41] (as mentioned in Sec. 1.1.1). In Grillner's model, each limb of an animal is controlled by a separate CPG. The CPGs are coupled through interneurons controlling the inter-limb coordination. The resulting controller is reconfigured to obtain gait transitions. Different gaits are controlled by different sets of coordinating interneurons. This approach is in principle biologically reasonable, but it has not yet been experimentally established that supraspinal centers recruit functionally distinct sets of coordinating interneurons to control different gaits. Stafford and Barnwell [108] adopted these ideas, modeling quadrupedal locomotion with a CPG that consisted of four coupled networks of oscillators, each of which controlled the muscles of one limb. By changing the coupling strengths between certain oscillators, they were able to make the model walk, trot and bound with gait transitions from walk to trot and from walk to bound.

The same three gaits can also be obtained with a network of four symmetrically
coupled oscillators without changing the coupling strengths between oscillators, but instead by changing either internal oscillator parameters or the network’s driving signal [22]. Similarly, Canavier et al. [18] obtained several quadrupedal gaits with a ring of four coupled oscillators of physiologically reasonable complexity [17]. They showed that gait transitions could be obtained by variation of the intrinsic properties of the circuit’s model neurons by changing the global stimulation intensity. However, changes in the patterns produced by a ring circuit can also be obtained by changing synaptic properties such as the synaptic strength and the synaptic time constant, as was shown for a ring of three model neurons [110] with intrinsic dynamics very similar to the ones studied in [18].

In summary, it is theoretically possible to model quadrupedal gait control with small networks of coupled oscillators. Typically, these networks consist of one oscillator per joint. However, under certain constraints (symmetric coupling between identical oscillators) it is necessary to use networks of eight oscillators to produce the frequently observed quadrupedal gaits. Theoretically, it is possible to change the patterns produced by these networks by variation of a global signal that changes either the intrinsic dynamics of all oscillators or the coupling strengths between all oscillators. Alternatively, it is also possible to change the patterns produced by these networks by changing the coupling strength between certain oscillators locally. Both ways of modeling changes in animals’ gaits are biologically plausible. Additional experimental data is needed concerning the functional organization and operation of locomotor CPGs to determine the actual biological implementation.

For a robotic application, it is of practical use to build the controller for inter-leg coordination such that a large variety of different gaits (including the ones observed in animals) can be obtained. It is advantageous to be able to change movement patterns either abruptly or gradually, depending on the situation. Furthermore, to build robotic controllers which are minimal solutions, i.e. using as few oscillators as possible to control the desired variety of walking gaits is advantageous because it keeps complexity, cost and power consumption of the controllers low. Therefore, the symmetrical networks discussed first by Collins, Stewart and later together with Golubitsky and Buono are not necessarily ideal, because for four oscillators, their output is restricted to a number of patterns too small to account for all frequently observed quadrupedal gaits (the exact number depends on the specific network architecture [23]). Scaling up the network to eight oscillators allows for more patterns, but they are restricted by the network symmetry, not always in desirable ways. For example, the ‘canter’ gait is modeled by the following phase relationships: with respect to the LF foot, the phase lag of the RF foot is given by \( \varphi \), the phase lag of the LH foot is 0.75 and the phase lag of the RH foot is \( \varphi - 0.25 \). The in-phase relationship between LF leg and RH leg (compare Fig. 1.1) requires that \( \varphi = 0.25 \). The resulting gait is only a crude approximation of a canter, in that it differs from the phase relationships of the canter as classified by Alexander [2] and used by Collins and Stewart [23].
For practical purposes, networks of very simple oscillators are used to control the locomotion of a robot in the present thesis. The behaviour of a single, isolated oscillator can be described by linear differential equations with two variables. Possible control architectures are constrained by the choice of the oscillator. These constraints will naturally lead to a minimal control architecture for robotic four-legged walking (see Chapter 2 and Chapter 3).

1.2 Previous work

1.2.1 Computer-controlled four-legged walking robots

The first autonomous four-legged walking machine [77] was built in the 1960's and was controlled by a finite state machine, which is a distributed control scheme [76]. The controller was implemented on a digital computer and the robot was programmed to execute a walk gait and, with a slightly modified controller, a trot gait. Since then, other (four and six-legged) computer controlled, statically stable robots have been implemented (for an overview, see [62]). The controllers of some of them were biologically inspired and used distributed controllers ([13] [92] and [93] [30]) or, more specifically, simulated networks of coupled oscillators ([7] [6]). It should be noted that considerable progress has been made in the field of dynamic leg control for robots (see [95] and [96]), but this subject is not a direct objective of the present thesis.

Some approaches should be emphasized. Control theory can be successfully applied to walking robots only if the joint torques (or the forces) which must be generated such that the machine produces a specified pattern of movement can be computed [120]. Computing the inverse kinematic equations to control a legged machine can be extremely CPU intensive [78]. Distributed motor control requires less computational power [30]. Subsumption architecture, introduced by Brooks in the late 80's (see e.g. [12] and [11]) makes use of this more biologically inspired principle of distributed control. Subsumption architecture decomposes the control problem into task-achieving layers. These layers are hierarchically organized. Lower layers run continually and are unaware of higher layers, but higher layers can choose to subsume the role of lower layers. The main idea is that layers of a control system can be built corresponding to each level of competence. By adding a new layer, a higher level of overall competence can be achieved. Within the scheme of subsumption architecture, three different biologically inspired models were implemented on a hexapod and tested with respect to flat and rough terrain locomotion capabilities and fault tolerance [32]. The implementation consisted of approximately 1500 concurrently running processes performing different tasks.

Another biologically inspired controller that is implementing a neural network control architecture on a six-legged machine with two degrees of freedom (DOF) per
leg was introduced by Beer and colleagues (see [6], [20] and [94]). The controller is composed of six identical sub-circuits, one for each leg. Each sub-circuit consists of an oscillating pacemaker neuron which, by inhibiting the neurons that make the leg execute a backward movement while supporting the body and by exciting the neurons that cause a forward swing, triggers the onset of a series of movements that result in the step cycle of the leg. Inter-leg coordination is achieved by mutual inhibition of the pacemaker neurons of adjacent legs. To switch between gaits, a central command neuron excites all of the pacemaker cells (thereby changing their frequency) and all of the neurons responsible for backward swing. This network produces a range of hexapod wave gaits and generates robust locomotion. If this network was reduced to a controller for four-legged inter-leg coordination, it would have the architecture of a ring of four symmetrically coupled cells. Applying the findings of Collins and Stewart [23], this symmetry implies that the resulting network could control some of, but not all of the standard quadruped gaits.

All of the robots mentioned above are carefully engineered machines and are controlled by one (or several) digital computer(s) running the algorithms that implement the respective controllers and thus they involve much machinery. Tilden [50] showed that legged robots can be controlled in much simpler ways. He approached the problem of legged robots and their control from a radically different angle, building imperfect, hand-made machines and controlling them with minimal controllers composed of a few discrete components. The behaviors of the resulting machines had a complexity that was remarkable given the simplicity of the controllers. Tilden's machines are more analogous to biological walkers in the sense that they use the physics of the units which make up the controller in a much more direct way. The controllers do not run any programs, but instead exploit the pattern generating capability of the hardware oscillators. First attempts to explain the behavior of these robots concentrated mainly on classification and description of the complexity of the robots' behaviors [52]. In this thesis, Tilden's approach is analyzed in terms of circuit analysis. The pattern generating capabilities of the most important oscillator networks used in his approach are studied analytically. As a result of this, the architecture of a minimal controller that can generate all of the standard quadrupedal gaits (as shown in Fig. 1.1) is determined (see Chapter 2). A shortcoming of the oscillator networks studied in Chapter 2 is that the patterns they generate depend on the resistances and capacitances of discrete components which can be changed mechanically, but not electronically. This can be improved by the use of transistors instead of resistors. The resulting circuits can be implemented on an aVLSI chip to yield a compact device (see Chapter 3).

1.2.2 Neuromorphic chips for motor control

Neuromorphic Engineering [79] deals with the development of Very Large Scale Integrated (VLSI) chips containing (mostly analog) electronic circuits that mimic neurons and neuronal circuits present in the nervous system. Neuromorphic compu-
tation seeks to use computational principles that can be elucidated from the nervous systems of animals. Neuromorphic systems typically directly exploit the physics of their hardware to execute their computations.

The first neuromorphic systems implemented sensory processing to some degree of complexity (e.g. [80], [75] and [74]). Since the field of neuromorphic engineering was pioneered by Mead in the 1980’s [114], considerable progress has been made on sensory systems, such as visual ([10], [60], [67], [66] and [73]) and auditory systems [34]. Sensorimotor systems were built to model integrative functions of the nervous system, such as the primate oculomotor system [57].

Ryckebusch et al. [101] introduced the first neuromorphic central pattern generator in 1989. The circuits they used to model CPG neurons (see Fig. 1.2) consisted of a pulse generating circuit (producing neuronal ‘spikes’) [79], and four synapses, two excitatory, two inhibitory. Each synapse was modeled with two transistors. The excitatory synapses consisted of two p-FETs¹ and the inhibitory synapses of

¹Field Effect Transistor (FET)
two n-FETs each. One of the transistors of each synapse was used to set the strength of the synapse while the other transistor implemented the synaptic input. The third element in the neuron's circuit implemented a temporal delay using a follower-integrator circuit [79]. Two synapses, one excitatory and one inhibitory, were located before the delay, modeling slow synapses. The others, undelayed synapses, modeled fast synapses. Networks of these model neurons could generate bursts of action potentials with phase relationships between the bursts. Two reciprocally connected neurons with a delay on only one of the connections were used to model the output of the 301 and 501 cells in the locust flight CPG. A network of three neurons was used to model the phase relationships of the outputs of neurons controlling the swimming movements of the marine mollusk Tritonia.

Ryckebusch went on to model rhythmic locomotor patterns with neurons of greater complexity that were more closely analogous to real neurons than her first CPG neuron model. She used model neuron circuits similar to a Hodgkin-Huxley sodium-potassium conductance pair and modeled the rhythmic motor patterns controlling leg movements in a locust (Schistocerca americana) [102]. In [102] the authors point out that their aVLSI implementation of a single leg model CPG could be used as a single leg controller in a walking robot.

Similar aVLSI model neurons have been interfaced to real biological neurons to study central pattern generator circuits in the stomatogastric ganglion of lobsters ([71], [113]).

Recently, another biologically realistic neuron model, based on a model by Morris and Lecar [83] was implemented in VLSI technology ([87] and [86]) and used to model the coordination of axial locomotion in swimming animals such as leeches and lampreys [85]. The control circuit consisted of a chain of twelve pattern generators each of which was composed of two silicon versions of Morris-Lecar model neurons. The pattern generators were capable of arbitrary contralateral inhibitory synaptic coupling. An asynchronous address-event interconnection element connected the model neurons and implemented axonal delay. The resulting controller had 384 programmable inhibitory synapses, sixteen per model neuron.

In contrast, in the model for inter-leg control described in this thesis, implements much simpler oscillators with fewer control parameters. The output waveforms of the controllers are abstractions of neuronal bursts, capturing only those values of the neuronal output which are relevant for controlling the inter-leg coordination of a robot. These are (i) the burst duration, (ii) the inter burst frequency and (iii) the phase lag between the onsets of bursts of different neurons. The equivalent values for the output waveforms of the controllers described in this thesis are (i) the duty cycles of each oscillator, (ii) the common frequency, and (iii) the phase lags between oscillators. These circuits are more compact than the original CPG neuron model by Ryckebusch et al. [101]. They are sufficient to control the inter-leg coordina-
tion of four-legged robots in that they can control all of the standard walking gaits. The square wave voltage outputs of these controllers can directly control dc-motors.

The work presented in this thesis is original in that it is to our knowledge the first neuromorphic chip for inter-leg control used on a real walking robot. It is based on a systematic analysis of a previously existing approach, and implements inter-leg control in a minimalistic way, using circuits which are as simple as possible. The control chip is tested on a four-legged robot with one DOF per leg (see Chapter 4).

Many robot designers either preprogrammed the leg sequence for a desired walking gait (see [56], [77] and [107]) or used CPG controllers and adjusted the parameters of the controllers such that they obtained the desired gait(s) [8]. Lewis et al. evolved controllers for six-legged walking using genetic algorithms, and used adaptive oscillators to model lamprey swimming and the walking behavior of a robotic model of a tetrapod. For a six-legged robot, they were able to evolve an efficient gait. The procedure involved an experimenter scoring the performance of the robot. The robotic tetrapod had a flexible spine and they showed that the same pattern of activity which could make a lamprey model swim could also establish a basic salamander-like walking gait in the tetrapod.

In contrast to this approach, which focused on evolving controllers for locomotion in a way similar to biological evolution, in this thesis the walking controller is developed based on analytical findings. But, since the VLSI chip is not a perfect device but rather subject to process dependent mismatches and offsets [88], the bias parameters which determine the gait that the robot uses must be fine-tuned. This is achieved here in an automated way that does not require the intervention of an experimenter. The problem is reduced to a classification task which is solved using an unsupervised Support Vector algorithm that was introduced recently [103].
Chapter 2

Walking gait control using coupled oscillators implemented with discrete components

This chapter deals with controllers for legged locomotion which are implemented by means of networks of coupled oscillators. Electronic circuits that have previously been used to generate walking behavior (see e.g. [50], [111] and Sec. 2.1) are analyzed (Sec. 2.2 and Sec. 2.3) and possible applications are explored (Sec. 2.5). The circuits are chosen because they are composed of very few components, yielding compact networks of coupled oscillators suitable for walking control. Furthermore, their output square wave voltage signals capture the essence of the output of bursting motor neurons. The variables which describe those aspects of the behavior of a bursting motor neuron which are important with respect to the coordination of the movements of motors used to drive the legs of a robot are the burst length, the inter burst frequency, and the phase lag between bursts of two different neurons. The burst length is captured by the duty cycle of the square wave. The inter burst frequency corresponds to the frequency of the square wave. Finally, a phase lag can be defined for two square wave outputs in the same way as it can be defined for two bursting neurons.

The aim of this chapter is to clarify which networks of coupled oscillators of this particular kind can be used in a robotic application to control the walking gaits of a four-legged machine, enabling the machine to use all of the frequently observed walking gaits of four-legged animals (which are sketched in Fig. 1.1). For this purpose, the circuit parameters which control the frequency and the duty cycle of an oscillator and those which control the phase lag between oscillators are identified. The way in which they control these variables is explored theoretically (Sec. 2.2.2 and Sec. 2.2.3). Based on this understanding, two different architectures of controllers are considered, namely rings (Sec. 2.3.1) and chains (Sec. 2.3.2) of oscillators. The patterns they can produce which are relevant to walking gait control are determined. Furthermore, a natural extension of the oscillator circuit is analyzed and the utility
of the resulting circuits with respect to walking gait control is discussed in Sec. 2.3.3.

Following from the preceding analysis, Sec. 2.4 describes how sensors can be used to change the control parameters, and thereby alter the robot's behavior. This enables the robot to show autonomous responses and simple behaviors (see [51]). Finally, applications of two different controllers to robotic legged locomotion are presented in Sec. 2.5.

2.1 Introduction

The circuits that are analyzed in this chapter originate from the work of M. W. Tilden (see e.g. [50], [52], [116] and [117]). In recent years, he has developed a large variety of autonomous, walking robots, the leg movements of which are controlled by networks of coupled oscillators implemented as electronic circuits with discrete components. Tilden uses hard wired sensory input to alter the controllers outputs. This results in walking machines whose behaviors can be remarkably complex compared to the simplicity of the oscillators.

Previous explanations given by Tilden and his collaborators have focussed on the qualitative description of the range of behaviors of the machines [50]. In contrast, this chapter provides a quantitative analysis of the controllers. The analysis yields an understanding of the controllers' outputs and that, in turn, provides the basis for understanding the behavior of the walking machines that are driven by the controllers.

The step frequency of the robot's legs is controlled by the oscillator frequency.

Figure 2.1: Examples of two controller architectures analyzed in this chapter. Arrows between oscillators indicate coupling.
Turning behavior results from a change in the duty cycle of a subset of the oscillators. The walking gait is determined by the phase lag between the oscillators. The circuit analysis clarifies the way in which the oscillator's frequency and duty cycle, and the phase lag between oscillators depend on the values of a small number of control parameters and how they can be changed by sensory input, yielding changes in the robot's behavior.

2.2 Analysis of the circuits

The aim of this chapter is to determine which patterns that are suitable for walking gait control can be generated by the electronic circuits. The electronic implementation of an oscillator by means of a few discrete components is introduced and the behavior of the oscillator is analyzed (Sec. 2.2.2). Each oscillator contains two sub-circuits which are characterized by individual time constants. The result of Sec. 2.2.2 is that the period of the oscillator is proportional to the sum of its time constants while the duty cycle is proportional to the ratio of one of the time constants to the sum of both time constants. These simple relationships make the oscillator very easy to control by adjusting the values of resistances or capacitances.

The oscillators discussed in Sec. 2.2.2 are coupled by resistors. The coupling effects the internal voltage of only one of the oscillators (the slave), while the other (the master) maintains its uncoupled oscillation frequency. The phase shift between two oscillators as a function of the resistances in the circuit is determined (Sec. 2.2.3). There are two cases to be distinguished: (i) the network of coupled oscillators contains a master oscillator, in which case the common frequency of the network oscillation is identical to that of the master independent of the coupling resistors (within some range of values), (ii) all oscillators in the network are slaves of other oscillators, in which case the common period depends on the coupling resistors. For the second case the resulting common oscillation period is calculated. Furthermore an expression for the duty cycle of the slave oscillators is derived. Altogether, the derived functions clarify how the gait and direction of walking robots can be altered.

2.2.1 Notation

The following notation is used throughout this chapter. The two sub-circuits (see Fig. 2.2 and Sec. 2.2.2) that constitute each oscillator (see Fig. 2.3 and Sec. 2.2.2) have the same architecture and are connected in a symmetrical way. The subscripts \( l \) and \( r \) (for “left” and “right”) are used to distinguish the variables of these two sub-circuits. Each sub-circuit is characterized by a voltage \( V \) at the input node of its inverter, the voltage at the output of its inverter, \( V_{\text{out}} \), and by the values of a capacitor \( C \), a resistor between \( V \) and ground, \( R \), and a coupling resistor between two oscillators \( i \) and \( j \), \( R_{c,i,j} \). For simplicity all capacitors are kept constant.
Figure 2.2: Threshold element used as part of the oscillator: A differentiator in series with an inverter. The circuit reacts to a positive step input at node $V_{in}$ with a low pulse of duration $T \sim RC$ at node $V_{out}$. Therefore, this element transfers an incoming pulse to an outgoing pulse (both are marked with *), and sets the duration $T$ of the outgoing pulse.

$(C = 0.22\mu F)$ unless otherwise stated. Each oscillator outputs two mirror-image step function voltages, $V_{out,l}$ and $V_{out,r}$ and has two internal input voltages $V_{in,l}$ and $V_{in,r}$. $V_{in,l} = V_{out,r}$ and $V_{in,r} = V_{out,l}$. The subscripts $(l,r)$ are used for the sub-circuits only if variables from both sub-circuits are involved in an equation. When $n$ oscillators are considered, each of them is given a subscript $j = 1, \ldots, n$. In equations which are true for all $j$ the respective subscripts are omitted.

Data

The data used throughout this chapter for validation of the model is drawn from experiments using off-the-shelf components. The resistors used have a tolerance of 5%; the capacitors used have a tolerance of 20%. Standard 74HCT240 Hex-Inverter chips were used. The measurements were done with a Tektronix TDS 420A oscilloscope.

2.2.2 The Oscillator

Each oscillator (Fig. 2.3) is composed of two threshold elements (Fig. 2.2) by connecting the input of one to the output of the other and vice versa. Each threshold element consists of a differentiator in series with an inverter. It responds to a positive voltage step (from 0V (ground) to the supply voltage $V_{dd}$) at its input node with an active low pulse of duration $T$ at the output node of the inverter. Let this low pulse be a signal with signal duration $T$, by definition.
Figure 2.3: Circuit diagram of a two-component oscillator, a structure that is called a 'Bicore' by Tilden [118]. It is composed of two similar sub-circuits (Fig. 2.2). The values of the components in both sub-circuits need not be identical. The oscillator produces mirror-image step functions at the output nodes $V_{out, l}$ and $V_{out, r}$. These outputs can drive a DC motor, causing it to move with a given frequency and duty cycle which depend on the value of the two time constants $\tau_l = R_l C_l$ and $\tau_r = R_r C_r$.

**Frequency and duty cycle of the oscillator**

The frequency and duty cycle of the oscillator follow from the signal durations of both of its threshold elements. The signal duration, $T$, is equivalent to the time that the voltage $V$ at the input node of the inverter is above the threshold voltage, $V_{th}$, of the inverter. To calculate $T$, let us determine the time it takes voltage $V$ to decay from its maximum value at $t = 0$ to the threshold voltage of the inverter at $t = T$. The differential equation for the voltage at node $V$ (see Fig. 2.2) is given by

$$\dot{V} + \frac{V}{\tau} = \dot{V}_{in}$$

with $\tau = RC$ being the time constant. Let us choose $t = 0$ such that the voltage at $V_{in}$ is constant ($V_{in} = V_{dd}$) during the decay time. Hence for $t \geq 0$

$$\dot{V} + \frac{V}{\tau} = 0$$
$$V(t = 0) = V_{max}$$

where $V_{max}$ is the maximal voltage ($V(t = 0) = V_{max}$) and can be determined experimentally. This equation is solved by

$$V(t) = V_{max}e^{-t/\tau}$$
To calculate the signal duration $T$, note that $V(T) = V_{th}$. We obtain

$$T = \lambda RC$$

with

$$\lambda = \ln \frac{V_{max}}{V_{th}}$$

$T$ is proportional to the time constant $RC$ and hence can be modified by changing $R$ or $C$, for example with sensory input (see Sec. 2.4). Eq. (2.2) is in agreement with the measured data (compare Fig. 2.4).

The differential equations for the voltages $V_l$ and $V_r$ of the two threshold elements that form the oscillator are both of the form of eq. (2.1). We can assume that the maximum voltages and the threshold voltages are identical for both elements since the inverters are on the same IC. The period, $P$ of the oscillation is given by the sum of the signal times of each element (see Fig. 2.5):

$$P = T_l + T_r = \lambda (R_l + R_r)C$$

The resistors $R_l$ and $R_r$ need not be identical. Let us define the negative duty cycle, $D_k$, with respect to $T_k$, $k \in \{l, r\}$ as the time that the voltage at $V_{out,k}$ is low,
Figure 2.5: Period of a two-component oscillator vs. the sum of its resistors: $R = R_l + R_r$. Line: theoretical prediction, asterisks: measured data. For this circuit $V_{th} = 1.4V; V_{dd} = 5V$ and $V_{max} = 5.8V$.

divided by the period of the oscillation. It is given by

$$D_k = \frac{T_k}{P} = \frac{R_k}{R_l + R_r}$$  \hspace{1cm} (2.5)

The frequency, $\nu$, of the oscillation is related to the oscillation period by

$$\nu = \frac{1}{P}$$  \hspace{1cm} (2.6)

In some cases it might be useful to have only one resistor which controls the oscillator's frequency. This is implemented in a similar oscillator circuit (Fig. 2.6) in which one resistor, $R$, connects $V_i$ and $V_r$. Let us assume that the input to the left half of the oscillator, $V_{in,l}$ has just switched from ground to $V_{dd}$ ($V_{in,l}(t < 0) = 0$, $V_{in,l}(t \geq 0) = V_{dd}$ and $V_i(t = 0) = V_{max}$). The differential equations for $V_i$ and $V_r$ are

$$\dot{V}_i + \frac{1}{\tau}V_i = \frac{1}{\tau}V_r$$

$$\dot{V}_r + \frac{1}{\tau}V_r = \frac{1}{\tau}V_i$$

$$V_i(t = 0) = V_{max}$$
Figure 2.6: Oscillator with one resistor connecting the input nodes of both inverters. Similar to the two-component oscillator shown in Fig. 2.3.

With $\tau = RC$. This set of equations is solved by

\[
\begin{align*}
V_l(t) &= V_{\text{max}} e^{-2t/\tau} \\
V_r(t) &= -V_{\text{max}} e^{-2t/\tau}
\end{align*}
\]

We obtain the time it takes voltage $V_l$ to decay from $V_{\text{max}}$ to the inverter threshold

\[T_l = \frac{1}{2} \lambda RC\]  

The circuit is symmetric and hence $T_r = T_l$. Thus, the oscillation period becomes (compare Fig. 2.7)

\[P = \lambda RC\]

The duty cycle of this oscillator is 50%, since $\frac{T_l}{P} = \frac{T_r}{P} = \frac{1}{2}$. It can be altered with sensors that break the symmetry of the oscillator, for example photo diodes, resulting in a simple device that turns towards or away from the light [119].

### 2.2.3 The Coupling

In the circuit shown in Fig. 2.8, two oscillators are coupled by resistors. Each of the outputs of one oscillator is coupled to one of the input nodes of the other oscillator's inverters. At the low frequencies we are concerned with in this application (ca. 0.5 Hz - 2 Hz) the coupling only affects the second oscillator, because the inverters of the first oscillator act as impedance buffers. This makes the coupling unidirectional.

For ease of reference, let us call the first oscillator a 'master' oscillator and the second a 'slave' oscillator. In Fig. 2.8, the subscript $i$ denotes the master oscillator, while the subscript $j$ denotes the 'slave' oscillator.
The coupling can be made bidirectional by coupling the slave oscillator’s output to the master oscillator’s input. We can think of the resulting circuit as a ring of two oscillators. The behavior of oscillator networks with a ring architecture is discussed in Sec. 2.3.1.

The coupling between the two oscillators causes a phase shift in the following way (see Fig. 2.9). When the voltage $V_{\text{out},i}$ steps from 0V to $V_{\text{dd}}$, charge is integrated at node $V_j$, therefore the voltage $V_j$ rises. After a delay time $T_D$, it reaches the inverter threshold and the voltage $V_{\text{out},j}$ switches from high to low, causing the other inverter of oscillator $j$ to switch from low to high $(V_{\text{in},j})$, which in turn causes voltage $V_j$ to jump to $V_{\text{max}}$. As a result, there is a phase shift $(\phi)$ between the output signals $V_{\text{out},i}$ and $V_{\text{out},j}$ of the two oscillators.

The frequency of the slave oscillator now depends both on the resistor connected to ground and on the coupling resistor. To calculate this frequency and the phase shift between two oscillators, we apply Kirchhoff’s current law at node $V_j$

$$\dot{V}_j + \frac{V_j}{\tau_{\text{eff},j}} = \frac{V_{\text{out},i}}{\tau_{c,j}} + \dot{V}_{\text{in},j}$$  \hspace{1cm} (2.10)

with

$$\tau_{\text{eff},j} = C R_{\text{eff},j} = C \frac{R_{c,j} R_i}{R_{c,j} + R_i}$$  \hspace{1cm} (2.11)
Figure 2.8: Two coupled oscillators. The coupling resistors $R_{c,j,r}$ and $R_{c,j,l}$ set the phase lag between the oscillators i and j. At low frequencies, the coupling is unidirectional because it influences only the voltages at the input nodes of the inverters of oscillator j, not those of oscillator i.
Figure 2.9: The integration of charge at node $V_{j,l}$ that occurs due to a step at $V_{out,l,l}$ causes the left inverter of oscillator $j$ to switch from high to low after a delay $T_{D,j}$, which in turn causes the right inverter of oscillator $j$ to switch from low to high ($V_{in,j,l}$). The signal times of the oscillators $i$ and $j$ are denoted by $T_{i,k}$ and $T_{j,k}$, respectively, with $k \in \{l, r\}$. One can see from these oscilloscope traces that the phase lag between the two output waveforms ($V_{out,i,l}$ and $V_{out,j,l}$), defined as the time between onsets divided by the period, is given by $\phi_l = (T_{D,i} + T_{j,i})/P = (T_{D,r} + T_{i,r})/P$. Note that $V_{out,j,l}$ is the mirror image of $V_{in,j,l}$, and the latter is displayed here. In the same way, the phase lag between $V_{out,i,r}$ and $V_{out,j,r}$ is given by exchanging $r$ and $l$. These two phase lags are different from each other if both oscillators do not have the same duty cycle.
We consider that part of the oscillation, during which $V_{out,i} = V_{dd}$. We are interested in two cases: (i) $V_{out,j}$ has just jumped from $0V$ to $V_{dd}$. We chose $t = 0$ to be just after the transient. When this happens, the voltage $V_j(t = 0)$ is close to $0V$ (compare Fig. 2.9). $V_{in,j}$ is constant and thus the term $V_{in,j}$ disappears. By solving the remaining equation and calculating how long it takes $V_j$ to rise to the threshold voltage, we obtain the delay time $T_D$. (ii) $V_{in,j}$ has just jumped from $0V$ to $V_{dd}$. Again, we chose $t = 0$ to be just after the transient, such that $V_{in,j}$ is constant for $t \geq 0$. The voltage $V_j(t = 0)$ is at $V_{max}$ while $V_{out,i} = V_{dd}$ (compare Fig. 2.9). Solving the equation for this initial condition and calculating the time it takes $V_j$ to decay back to the inverter threshold leads to the signal time $T$ on either side of the slave circuit which in turn leads to the period of the slave oscillator.

Note that the inverter does not switch at the same voltage, depending on whether the voltage on the input of the inverter rises or falls [115]. This can clearly be observed in Fig. 2.9. In Sec. 2.2.2, we called the threshold voltage at which the inverter switches when the input to the inverter decays from a high to a low voltage, $V_{th}$. Let the voltage at which the inverter switches when the input to the inverter rises from a low to a high voltage be called $V_{th'}$.

Formally, we have to solve the same equation for both cases

\[ V_j(t) = V_0 e^{-t/\tau_{eff,j}} + \frac{V_{dd}}{r_j + 1}(1 - e^{-t/\tau_{eff,j}}) \]  

(2.14)

with

\[ r_j = \frac{R_{c,j}}{R_j} \]  

(2.15)

**Frequency of a slave oscillator**

To calculate the frequency of a slave oscillator, we consider eq. (2.14) with $V_0 = V_{max}$. To calculate the time $T_j$ it takes $V_j$ to decay from $V_{max}$ to the inverter
Figure 2.10: Time delay $T_D$ between two oscillators (see Fig. 2.9). The points are the measured data, the line corresponds to eq. (2.19). $V_{th}=1.49 \, \text{V}$, $V_{dd}=5\, \text{V}$.

threshold, we substitute $V_i(T_i) = V_{th}$, and obtain

$$T_j = C \frac{R_{c,j}}{r_j + 1} \ln \left[ \frac{V_{max}(r_j + 1) - V_{dd}}{V_{th}(r_j + 1) - V_{dd}} \right]$$

This equation is valid both on the left and right sides of the circuit. The period $P$ of the oscillator is once again given by $T_{j,l} + T_{j,r}$. To obtain the period for slave oscillators subscripts $l$ and $r$, respectively, have to inserted into eq. (2.16). They are omitted in the equation, because it holds for both sides of the circuit. The period is then given by

$$P_j = C \left( \frac{R_{c,j,l}}{r_{j,l} + 1} \ln \left[ \frac{V_{max}(r_{j,l} + 1) - V_{dd}}{V_{th}(r_{j,l} + 1) - V_{dd}} \right] + \frac{R_{c,j,r}}{r_{j,r} + 1} \ln \left[ \frac{V_{max}(r_{j,r} + 1) - V_{dd}}{V_{th}(r_{j,r} + 1) - V_{dd}} \right] \right)$$

Where $r_{j,k} = R_{c,j,k}/R_{j,k}$ with $k \in \{l, r\}$ (compare eq. (2.15)).

**Phase lag between two oscillators**

The phase shift between two oscillators is defined as the time difference between the onset of both oscillations divided by the period, assuming that both oscillations have the common period $P$. In general, the parameter values of right and left sides of the oscillators are not identical. Let us define two phase lags, one for each half
of the circuit. They are labeled with \( r \) and \( l \), and are given by

\[
\phi_r = \frac{T_{D,r} + T_{j,r}}{P} = \frac{T_{D,l} + T_{i,l}}{P}
\]

and similarly for \( \phi_l \), with \( l \) and \( r \) exchanged (see Fig. 2.9). \( T_{i,k}, T_{j,k} \) (\( k \in \{l, r\} \)) and \( P \) are known from the proceeding analysis. Note that we have to distinguish if there is a master oscillator in the circuit or not. If there is one, it drives the slave oscillators and the common period is equal to the period of the master oscillator (see eq. (2.4)). If all oscillators are slave oscillators, they must have a common period according to eq. (2.17).

To calculate the delay times \( T_{D,l} \) and \( T_{D,r} \), we consider eq. (2.14) with \( V_0 = 0 \), substitute \( V_j(T_{D,j}) = V_{th} \) and solve for \( T_{D,j} \) (compare Fig. 2.10)

\[
T_{D,j} = C \frac{R_{c,j}}{r_j + 1} \ln \frac{V_{dd}}{V_{dd} - V_{th}(r_j + 1)}
\]

This equation holds for both sides of the circuit and thus the subscripts \( l \) and \( r \) are omitted.

**Implications for Gait Control**

If each oscillator drives the rhythmic movement of one of the legs of the robot, then different walking gaits are distinguished by the phase lags between the oscillators. Eq. (2.18), together with eq. (2.19) clarifies how the phase lags can be controlled by the values of the coupling resistors. Depending on the presence or absence of a master oscillator in the network, the common period is given by eqs. (2.4) or (2.17), respectively, while the signal duration is given by eqs. 2.2 and 2.16, respectively. We make use of these two different cases in the following sections when we determine the patterns (and with them the gaits) that chains or rings of oscillators can support.

The direction of the robot can be altered by introducing a left-right asymmetry in the forces that are exerted upon the robot’s body by the movement of its legs. This can be done by changing the duty cycle of the oscillators (see Sec. 2.5 and Sec. 4.3.2 for a more detailed explanation). The more asymmetric the duty cycle becomes, the stronger the turning effect. Hence, knowing how to control the duty cycle (Eqs. (2.5) with (2.2) and (2.16), respectively) enables us to control the direction of the robot.

### 2.3 Patterns produced by oscillator networks

Based on the analysis of the previous section, we can now explore the range of patterns relevant to four-legged locomotion that are supported by networks of oscillators. The advantages and disadvantages of controllers with different architec-
tures - rings and chains of oscillators - are highlighted. In particular it is shown how the control over the walking machine can be facilitated considerably by using a chain of oscillators in which the slave oscillators do not contain resistors to ground. With this controller, the control over the walking gait and the direction the robot is decoupled from the control over the step frequency of the legs. The phase lag between oscillators becomes proportional to the ratio of the coupling resistance and the resistance of the master oscillator.

A natural extension of the oscillator circuit described in Sec. 2.2.2 is analyzed in Sec. 2.3.3. Each oscillator is composed of two identical sub-circuits that are reciprocally connected to one another. This structure is a ring of the two sub-circuits. Rings of larger numbers of these sub-circuits can have multiple modes and hence support several different walking patterns. The number of possible modes for rings of more than two sub-circuits is determined and the limitations of these circuits for a gait control application are discussed.

### 2.3.1 Rings of oscillators

As mentioned before, the neuronal generation of animal gaits is frequently modeled with rings of n coupled oscillators, where n is typically the number of legs (see Sec. 1.1).

To investigate which kind of patterns can be produced by rings of the oscillators described here, let us look back at eq. (2.14). In the limit \( t \rightarrow \infty \), \( V_j(t) \) approaches \( \frac{V_{dd}}{r_j + 1} \), independent of the value of \( V_0 \). For a ring of oscillators to be able to oscillate without an external driving signal, the value of \( \frac{V_{dd}}{r_j + 1} \) has to lie between \( V_{th'} \) and \( V_{th} \). If it would be below \( V_{th'} \), the voltage \( V_j(t) \) could never cross the inverter threshold after a rising edge at \( V_{out,i} \), which would mean that the oscillation would stop. If the voltage \( \frac{V_{dd}}{r_j + 1} \) would be above \( V_{th} \), the voltage \( V_j(t) \) could never decay back down below the inverter threshold after a rising edge at \( V_{in,j} \), which would mean that oscillator \( j \) would need the output of oscillator \( i \) as a driving signal. This constraints the ratio between coupling resistance and slave resistance to be

\[
\frac{V_{dd}}{V_{th}} - 1 < r_j < \frac{V_{dd}}{V_{th'}} - 1
\]  

(2.20)

To estimate this range, we substitute the values of Fig. 2.9 (\( V_{dd} = 5V \), \( V_{th} = 1.6V \) and \( V_{th'} = 2.4V \), estimates, not precise values) and obtain (given to one decimal place)

\[
1.1 < r_j < 2.1
\]  

(2.21)
Rings of coupled, identical oscillators

One of the requirements for a walking gait controller is that it can generate forward walking. To do so, the networks considered here must be able to generate patterns with identical 50% duty cycles. This behavior can be obtained by giving the parameters on the left and the right side of the circuit identical values. For the control of some walking gaits, as the walk gait and the trot gait, the phase lags between the oscillators have to be identical. This arrangement can be implemented using identical oscillators (including identical coupling). For identical oscillators with identical parameter values on both sides, the delay times and the signal times are identical \((T_{D,i,l} = T_{D,i,r} \equiv T_D \text{ and } T_{i,l} = T_{i,r} \equiv T ; \ \forall i)\). The phase shifts between the oscillators are then given by \(\phi_{i,l} = \phi_{i,r} \equiv \phi ; \ \forall i\), with

\[
\phi = \frac{T_D}{T} + \frac{1}{2}
\]

(2.22)

By definition

\[
\frac{1}{2} \leq \phi \leq 1
\]

(2.23)

which is equivalent to saying that \(T_D \leq T\). In rings of oscillators, each oscillator must be in phase with itself. Hence the phase lags of the \(n\) oscillators must sum up to an integer \(z\) \((z \in \mathbb{N})\). Since the phase lags are all identical, this condition reads

\[
\phi = \frac{z}{n}
\]

(2.24)

Substitution of eq. (2.24) into eq. (2.23) yields

\[
\frac{n}{2} \leq z \leq n
\]

(2.25)

Substitution of (2.22) into (2.24) yields

\[
nT_D = (2z - n)T
\]

(2.26)

Thus, to obtain a phase shift of 0.5 (with \(z = n/2\)), \(T_D\) must be zero. This can be implemented by choosing the value of the coupling resistance to be zero. For all other values of \(z\), the value of \(\tau\) that corresponds to a pattern with \(\phi = z/n\) can be obtained as a numerical solution of

\[
\ln \left[ \frac{V_{dd}}{V_{dd} - V_{th}(r_j + 1)} \right] = \frac{2z}{n} - 1
\]

(2.27)
Left front (LF)  Right front (RF)  Left hind (LH)  Right hind (RH)  oscillator

Figure 2.11: Architecture of a walking controller consisting of 8 coupled oscillators.

Table 2.1: Gaits that a ring of 8 oscillators can produce. Phase lag, corresponding gait and value of $r = R_{c,j}/R_j$

| $\phi$ | 1/2 | 5/8 | 3/4 | 7/8 | 0 |
| Gait   | pronk | transverse walk | bounce | walk | pronk |

Examples

A ring of 4 oscillators can produce three patterns. If the controller is connected to the robot’s legs in the way shown in Fig. 2.1, the pattern with $\phi = 1/2$ corresponds to a bound. The pattern with phase lags of 3/4, corresponds to a walk gait and that with $\phi = 1$ corresponds to a pronk.

A ring of 8 oscillators can produce five patterns, which are listed in Table 2.1. The corresponding gaits are written down for an architecture like the one in Fig. 2.11.

Discussion

The advantage of controllers with a ring architecture is that one global control parameter, applied to all oscillators, changes the gait. This can be done, e.g., by keeping $R$ constant and changing only $R_c$. At the same time this changes the oscillator frequency and this in turn changes the step frequency of the robot’s legs. The latter feature can be seen as an advantage, because it means that at a different speed the robot automatically chooses a different gait. However, this is only an
advantage if the gaits become increasingly more efficient for higher speeds. For the oscillators discussed here, this is not necessarily the case as can be seen in the case of a four oscillator network, where we have the transitions: bound - walk - pronk. Furthermore, both a ring of four identical oscillators and a ring of eight identical oscillators are not able to produce all of the most prominent gaits that four-legged animals use. For that, non-identical oscillators would have to be used, to yield non-identical phase shifts between oscillators. Then, there would be more than one control parameter for the circuit.

For many real world applications, it is an advantage to be able to control speed and gait independently. Let us examine an alternative circuit with a small number of oscillators and a different architecture: a chain of four oscillators.

2.3.2 Chains of oscillators

Chains of $n$ coupled oscillators have one master oscillator at the head of the chain, the period of which follows eq. (2.4). To simplify the control over the system, one can remove the resistors to ground in all oscillators except for the master oscillator. Then, the frequency of the system is solely set by the value of the master resistors (see eq. (2.4)). The delay time between the two oscillators $j - 1$ and $j$, $T_{D,j}$, depends only on the coupling resistances. If the ratio of the two coupling resistances contained in one oscillator ($R_{c,j-1,j}/R_{c,j-1,j,p}$) differs from 1, the duty cycle of this oscillator and of all oscillators that come below it in the hierarchy of the chain are affected. To determine how the phase lag and the duty cycle depend on the coupling resistors for this modified circuit, we apply Kirchhoff’s current law at node $V_j$

$$\dot{V}_j + \frac{V_j}{\tau_{c,j}} = \frac{V_{dd}}{\tau_{c,j}}$$

$$V(t = 0) = 0$$

with the solution

$$V_j(t) = V_{dd}(1 - e^{-t/\tau_{c,j}})$$

(2.28)

The delay $T_{D,j}$ once more is obtained by calculating the time it takes $V_j$ to rise to $V_{th}$. It is now simply proportional to the coupling resistance.

$$T_{D,j} = R_{c,j} C \ln \frac{V_{dd}}{V_{dd} - V_{th}}$$

(2.29)

$T_{D,j} < T_{master}(= T_1)$ is true for

$$R_{c,j,l} \leq R_{m,l} \frac{\ln (V_{max}/V_{th})}{\ln (V_{dd}/(V_{dd} - V_{th}))}$$

(2.30)
Figure 2.12: Period of a slave oscillator divided by the period of the corresponding master oscillator as a function of the sum of the coupling resistances divided by the sum of the resistances to ground. The circuit is of the type shown in Fig. 2.8. $V_{th} = 2\, V;\, V_{max} = 5.4\, V$ and $V_{dd} = 5\, V$.

Where $R_m = R_{i1}$ is the resistance of the master oscillator, which is oscillator 1 of the chain.

With typical values of the constants, $R_{c,j,l} \leq 1.79\, R_{m,i}$. If the coupling resistance is larger than this, the period of the slave oscillator becomes larger than the period of the master oscillator (see Fig. 2.12).

There is no further restriction for chain architectures. The phase lags between oscillators depend linearly on the coupling resistors and can individually be set to values between 0.5 and 1. Hence the control of patterns is significantly simplified.

To control turning one needs to know how to change the duty cycle of individual oscillators in the chain. The negative duty cycle with respect to $T_{r,j}$ of oscillator $j$ is given by

$$D_r = \frac{T_{D,j,l} - T_{D,j,r}}{T_{i,l} + T_{i,r}} + D_{i,r} \quad (2.31)$$

The subscripts $i$ stand for the oscillator that proceeds oscillator $j$ in the hierarchy of the chain ($D_i$ is given similarly with $r$ and $l$ exchanged).
Discussion
Controllers with chain architectures enable the robot to make transitions between different patterns smoothly (by changing the relevant resistances smoothly, for example, using a potentiometer) or abruptly (by switching between resistors). Smooth changes can be very useful in situations in which the change in the behavior of the machine should be gradual. This can be necessary for example for careful turning. Abrupt changes on the other hand are desirable for rapidly changing from one gait to another.

2.3.3 Rings of coupled threshold elements
Another way to create a circuit that is capable of producing multiple modes is to connect any number $n$ ($n \in \mathbb{N}$) of the threshold elements introduced in Sec. 2.2.2 (see Fig. 2.2) to form a ring. This is done by connecting the output of one threshold element to the input of the next threshold element. Let us concentrate on even numbers, because we want to use the circuits as walking controllers for animals with an even number of legs without breaking the symmetry in an unnatural way.

In a ring of connected threshold element of the kind shown in Fig. 2.2 the following is true:

- At the time when one threshold element in the ring switches from low to high, the next one switches from high to low.
- If undisturbed, a threshold element switches from low to high after a time set by its time constant, given by eq. (2.2).
- If an element that puts out a low pulse is disturbed by an input step in voltage from high to low, it switches back to high immediately.

The third rule implies that two adjacent ring elements can not output a low signal at the same time. The low output of the first threshold element is the input node to the second threshold element. If it is low, the input node of the inverter will also be low and hence the output of the inverter must be high.

For $n = 2$ the resulting ring is identical to the two component oscillator described in Sec. 2.2.2. For $n = 4$ the resulting controller has been implemented and is discussed in further detail in [111] (see Chapter 2.5.5).

Generally, the number of possible oscillatory patterns that can be produced by these ring networks increases with the size of the ring and can become quite large. Different patterns, or modes, arise from different initial conditions. To illustrate this, let us assume that all of the oscillators are in the resting state, in which the outputs of all inverters are high. Now, by forcing a number $z \in \mathbb{N}$ of elements to switch simultaneously from high to low the initial condition is set for a mode with
Table 2.2 lists the values of $M$ for even numbers of $n \leq 20$. This result is contrary to the one given in [50], which states that for $n = 10$, $M = 16$, for $n = 12$, $M = 32$, and for $n = 16$, $M = 128$.

Despite the large number of different patterns that they can generate, there are difficulties encountered when these ring controllers are connected to the robot’s
Figure 2.13: Wiring methods for a controller composed of 8 threshold elements, connected to a ring, driving a four-legged machine.

legs. The problem is to connect them such that the different modes produce reasonable walking gaits. This makes these ring networks less straightforward to use and therefore less attractive than the chains of two-component oscillators, considered before.

Each leg should move forwards and backwards within one cycle, and thus it is not possible to connect one of the inputs of the leg’s motor to the power supply and the other to a single oscillating ring element, because then the motor would only move in one direction. Instead, the motor must be connected between the outputs of two ring elements, one on each side.

Controlling a four-legged machine with a ring of four threshold elements is pos-
sible. In Sec. 2.5 a four-legged machine with 2 motors is controlled this way. The
variety of the walking gaits of this machine is limited both by the small number
of DOFs and by the small number of different modes of the controller (compare
Table 2.2). One question is whether extension of the ring yields a better controller.

To control a $n$-legged machine with 1 DOF per leg, we need at least $2n$ threshold elements to avoid that 2 legs have to share control elements, which would be a disadvantage, because it would mean that changing the duty cycle of one of the legs would alter the duty cycle of the other. This would place an undesirable restriction on the possible movements of the robot. Therefore, controlling a four-legged robot requires a ring of at least eight threshold elements.

Using a ring of eight threshold elements to control a four-legged walker with the constraint that a control signal drives no more than one leg, there are the following reasonable ways of connecting the legs to the outputs of the threshold elements: (i) the motor which drives the leg is connected to two adjacent ring elements (method 1). (ii) The leg is connected to elements that have one (method 2) or three (method 3) other elements between them. Leaving two elements between both sides of each motor is not possible. Fig. 2.13 illustrates the different wiring strategies.

If we denote one movement direction of a single leg with 1 and the other with 2, we can write down which leg moves in which direction at a given time of the movement sequence. For simplicity, let us assume that the threshold elements output signals of identical duration. Then the movement can be decomposed into eight distinct time steps. The resting state (11111111) results in no activity in each of the three cases, since there is no voltage drop across any of the motors. Tables 2.3 to 2.5 list the movement sequences for the remaining oscillatory patterns. Note that the direction of 1 and 2 can be chosen to be forward and backward or vice versa for each leg individually. The exact movement pattern which is caused by the activation sequences of tables 2.3 to 2.5 depends on that choice. If, as an example, we chose direction 1 = leg moves forward, and direction 2 = leg moves backwards, then we obtain the movement patterns listed in Table 2.6. This list shows that the overall performance is rather poor compared to the performance of controllers with two component oscillators and chain architecture which were discussed in Sec. 2.3.2.

In conclusion, the disadvantage of these controllers is that for rings with $n$ smaller than the number of legs, the possible movements are restricted by the fact that the legs share control signals, while for rings with $n$ greater than or equal to the number of legs, the wiring between the controller and the motors which determines the gaits is not as straightforward as in the case of chains of coupled oscillators. Furthermore we have seen by means of the example of controlling a four-legged machine with a ring of 8 threshold elements, there is no wiring method which provides that all patterns result in a sensible gait. This situation does not become better if one
<table>
<thead>
<tr>
<th>pattern</th>
<th>1: 01111111</th>
<th>2: 01011111</th>
<th>3: 01010111</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method 1</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>time step</td>
<td>LF</td>
<td>LH</td>
<td>RF</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Method 2</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>time step</td>
<td>LF</td>
<td>LH</td>
<td>RF</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Method 3</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>time step</td>
<td>LF</td>
<td>LH</td>
<td>RF</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2.3: Three different oscillatory patterns lead to different movement sequences when the controller is wired to the robot using the three different methods which are shown in Fig. 2.13.
<table>
<thead>
<tr>
<th>pattern</th>
<th>4: 01010101</th>
<th>5: 01101111</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Method 1</td>
<td></td>
</tr>
<tr>
<td>time step</td>
<td>LF</td>
<td>LH</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Method 2</td>
<td></td>
</tr>
<tr>
<td>time step</td>
<td>LF</td>
<td>LH</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Method 3</td>
<td></td>
</tr>
<tr>
<td>time step</td>
<td>LF</td>
<td>LH</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2.4: Continuation of Table 2.3
Table 2.5: Continuation of Table 2.4
Table 2.6: Gaits for the three different wiring methods of Fig. 2.13. S = all motors stall at all times. - = motors move but the resulting movement is not efficient in moving the robot in any direction. * = the robot moves (more or less), but the walking gait does not correspond to any of the ones listed in Fig. 1.1.

<table>
<thead>
<tr>
<th>Method 1</th>
<th>pattern</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>gait</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>pronk</td>
<td>*</td>
<td>-</td>
<td>*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method 2</th>
<th>pattern</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>gait</td>
<td></td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>S</td>
<td>-</td>
<td>*</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method 3</th>
<th>pattern</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>gait</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>S</td>
<td>walk</td>
<td>S</td>
<td>*</td>
</tr>
</tbody>
</table>

uses more threshold elements, because then for most patterns the legs will stall during even larger fractions of the cycle, resulting in inefficient or unnatural walking gaits. Therefore, we suggest not to use large rings of coupled threshold elements as controllers for robots with at least 1 DOF per leg. Instead, in the remainder of this thesis, chains of coupled oscillators are used which have the advantage that they can control different walking gaits easily with only a few control parameters (see Sec. 2.3.2). For the inter-leg control of a four-legged robot, for example, eight threshold elements are sufficient to control all possible walking gaits if the threshold elements compose a chain of four coupled two-component oscillators.

### 2.4 Sensory modulatory input

The walking machine can obtain information about the outside world with very simple sensors that can be integrated into the controllers described above. These sensors can be tactile sensors, for example. Antennae of different shapes were used. Their bases are located inside of tubes or attached to springs with contacts inside ([51]). When contacting an object somewhere along their length, they are bent and therefore shorten the contact at their base. The tactile sensors act as switches which can change resistances or capacitances in the controller and by doing so they change time constants and thus signal times, adjusting the duty cycle and the frequency of the oscillators to the encountered environment. They can provide information about the local or the distal environment depending on the length of the antennae. Distal sensory information is delayed before it changes the behavior of the controller.

The sensors can change one or several time constants in the circuits by changing one or several resistances or capacitances. By doing so they change the signal time of one or several threshold element(s) (see eq. (2.2)). If the resistances on
both sides of an oscillator are changed such that the ratio of the resistances remains constant, the frequency of the oscillator changes (see eq. (2.4)) but the duty cycle remains the same (see eq. (2.5)). If the ratio of the two resistances of an oscillator is changed by the activation of the sensor, the duty cycle of that oscillator changes. This can be used to make the robot turn, because it causes the leg movement which is caused by the turning of the motor to be asymmetric. The robot can be built such that as a result of this the applied force is different on both sides of the body. Furthermore, the delay time between two oscillators can be changed by sensory activation (see eq. (2.19)), resulting in a change of the phase shift between the oscillators and thus in a different movement pattern.

Any sensor which changes a resistance or a capacitance can be used in this way. Some of the simplest examples of these sensors are photo diodes, tactile pressure sensors, strain gauges, accelerometers and simple acoustic pressure sensors. More elaborate sensors can be used provided that their output can be converted into a resistance or a capacitance.

Local information is used to change a time constant immediately. An example for that is given in Sec. 2.5.

Information about objects that are not in the direct vicinity of the machine does not need to have an impact on the robot's immediate behavior. Rather, it should influence the robot's behavior in the near future. This can be realized by feeding the sensory signal into the controller via an integrating threshold element (Fig. 2.14, [50]), which delays the sensory signal for a time $T$ that can be adjusted with the
time constant $RC$ of the integrator.

$$T = RC \ln \frac{V_{dd}}{V_{dd} - V_{th}}$$

(2.33)

2.5 Applications

The smallest controllers of both kinds of architectures discussed above (rings and chains) were implemented to control a four-legged walking robot. In both controllers the number of threshold elements is four. They are arranged either as a ring (Sec. 2.5.3) or as a chain of two coupled two-component oscillators (Sec. 2.5.5). Both controllers can drive the robot. The differences between the controllers are discussed. Sections 2.5.1, 2.5.2, 2.5.5 and 2.5.6 are published in [111]. The contents of sections 2.5.3 and 2.5.4 are based on work published in [112].

2.5.1 The Robot and its Sensors

The robot consists of a flat, rectangular, 9cm x 3cm body, parallel to the ground, with 2 DC motors attached, as sketched in Fig. 2.15. Each motor drives a pair of legs, shaped in an inverted U with additional bends splayed away from the body of the robot for stability. The front legs are driven by a motor with its axis perpendicular to the body and can therefore move horizontally (drive), while the hind legs are actuated by a motor parallel to the body and move vertically (lift). As the hind motor turns, the legs attached to it stay on the ground so that the body rolls from side to side. This results in changing the weight distribution on the front legs. The sensors are short antennae attached to the front legs that act as local tactile sensors, while two photo diodes at the front of the body serve as distal optical sensors.

2.5.2 Data Acquisition

To quantitatively characterize the robot's behavior, the trajectories of the robot's feet are tracked and simultaneously the internal state of the controller is recorded. Light emitting diodes (LEDs) are attached to the robot's left front and left hind feet. A CCD-camera monitors the robot from one side. The bright LEDs are tracked by a custom made algorithm. This procedure gives the x and y positions of the feet as a function of time. The gait properties of the robot can be determined by observation of these traces. The internal state is given by voltages at several nodes of the controller.
Figure 2.15: Schematic of the robot. The upper drawings show a top and a side view, the lower one is a three dimensional sketch of the machine. LED = light emitting diode. The LEDs are used to monitor the movement of the robot’s feet as a function of time.
2.5.3 Walking controller made from a chain of two coupled oscillators

The controller (Fig. 2.16) consists of two coupled oscillators, one of which acts as a 'master' oscillator. They drive one motor each. Their period, duty cycle and the phase shift between them are given by eqs. (2.4), (2.5) and (2.18), respectively, and can be altered directly with sensory input through a change of the values of the resistances, as described in Sec. 2.4.

All sensory input can be used to change the four resistances of the controller, which in turn results in a range of behavioral responses, such as turning or change of gait. Here, the ratio of the values of the two master oscillator resistors $R_1$ and $R_2$ controls the ratio of lift duration between right and left front leg. For example, if $R_1/R_2 > 1$, the robot turns left, and conversely for right. If an object is detected on one side of the machine, the power stroke on that side is lengthened, causing the machine to turn away from the object. Changing the ratio between these resistances and the coupling-resistances ($R_3$, $R_4$), changes the delay between motor...
sweep paths and thereby the walking gait. By adjusting these resistances, the phase shift between front and hind girdle can be controlled.

Figure 2.17: Leg trajectories and internal voltages of the walking robot. See text for explanation (Sec. 2.5.4).

### 2.5.4 Locomotion of the Walking Machine

Figure 2.17 shows the voltages and leg trajectories obtained from the moving robot. The upper trace shows the $x$-position of the left front foot as a function of time ($x$-direction is in the direction of motion of the robot), the second trace shows the vertical position, $y$, of the left front foot, and the third and fourth traces show horizontal and vertical position of the left hind foot. The traces in the lower half of the figure are the simultaneously recorded internal voltages (notation corresponding to Fig. 2.16). The top trace shows the input to inverter 1. The second trace
shows the output of this inverter. The third trace shows the voltage at the input node of inverter 3, and the last trace shows the output of inverter 3. These four traces characterize the internal state of the robot. The first two traces show how the voltage $V_2$ decays proportional to $e^{-t/\tau}$ with $\tau = C \cdot R_1$, and triggers the oscillation of the 'master' oscillator by causing inverter 1 to switch when $V_2$ crosses the inverter threshold. The last two traces show how the coupling gives rise to the delay between 'master' and 'slave' oscillator. The outputs $V_{3\text{out}}$ and $V_{4\text{out}}$ are the mirror images of $V_{1\text{out}}$ and $V_{2\text{out}}$, respectively, and it would be redundant to monitor them. The outputs of the controller move the robot's feet in the following way: The turning of the hind motor (controlled by $V_{1\text{out}}$) results in a change of the height ($y$-position) of the front foot ($y_1$), while the hind foot drags on the ground ($y_2$).

As the machine steps, its limbs move the robots center of gravity from the back feet to the front ones. The robot does not so much step forward as 'fall' at plus and minus 45 degree angles to the forward direction of the robots' keel, as the motors cycle between the four step states.

The phase shift between front and hind feet is roughly 90 degrees ($x_1$ and $x_2$) in accordance with the phase shift of the controller, set to roughly 90 degrees by the ratio of the coupling resistances to the 'masters' resistances. The resulting gait resembles a walk gait. The speed of the robot is chosen to be low for ease of observation and is about 0.3 cm/s. The noise in the data is mainly due to the terrain which is not perfectly smooth.

### 2.5.5 Walking controller made from a ring of four threshold elements

The controller consists of four elements of the kind shown in Fig. 2.2, coupled together in a ring, as shown in Fig. 2.18. Each of these elements gives a low pulse output of duration $T$ as a result of an edge (abrupt rise from 0V to 5V occurring at its input node $V_{\text{in}}$). The signal duration time is given by $T = RC \ln \frac{V_{\text{max}}}{V_{\text{th}}}$ where $V_{\text{th}}$ is the threshold voltage of the inverter (see eq. (2.2)).

The outputs of these oscillating elements are used to drive the motors. The outputs of two of the elements that form the ring of the controller drive the front motor, and the other two drive the hind motor (see Fig. 2.18). The outputs of two elements are connected (via buffers) to the inputs of a motor and the voltage difference across the motor causes it to turn. When the voltage difference is reversed, the direction in which the motor turns is also reversed. The control signals thus cause the motors, and with them the legs, to move back and forth in a pattern that is determined both by the signal durations of the outputs as well as by the state (or mode) of the controller. As shown in Table 2.2, the controller can have 3 different states. In the resting state, no pulse travels through the ring and the robot does not move.
In the two other states there are either one or two pulses propagating through the ring, as shown in Fig. 2.18. These patterns gives rise to a walking motion. Since the legs of one girdle are attached to each other, the phase between them is fixed at $180^\circ$. The two possible modes of the controller yield two different gaits (see Fig. 2.18 to Fig. 2.20). Asynchronicity of the outputs of the elements in the ring, induced by non-identical time constants, can be used to induce turning behavior. The time constants of the elements can be individually modified in real-time by simple sensors that change a resistance.

2.5.6 Locomotion of the Walking Machine

The outputs that drive the motors are measured as internal states of the controller, to show the causal relationship between these voltages and the foot trajectories.

Figures 2.19 and 2.20 show the data recorded from the operating robot. The upper trace of each of Figs. 2.19 and 2.20 shows the horizontal displacement of the left front foot (positive displacement is in the direction of motion of the robot) as a function of time, the second trace shows the vertical position of this foot (where positive displacement means lift), and the third and fourth traces show the horizontal and vertical positions of the hind foot. The traces in the lower half of each figure show the voltage outputs of the oscillators that drive the motors. The upper two of these voltage traces drive the front motor, the lower two the hind motor.

In Fig. 2.19 the controller is in the state shown in Fig. 2.18 D. The vertical lines in the enlarged part of the figure are intended to indicate the relevant time intervals of a cycle.

During one half cycle of the oscillation the rear motor turns the body into a position that takes weight off the left front leg enabling it to move forward because of the simultaneous movement of the front motor. During this time the front foot moves forward (first trace). During the other half cycle weight is simultaneously put onto the left front foot by the movement of the hind motor and the foot is also moved backwards by the front motor. The hind foot moves forward in a rather continuous fashion, showing slight backward movements due to slipping.

The vertical displacement of both observed feet is negligible; variations are mainly due to the not ideally smooth floor. This means that the shift in weight between the front legs, induced by the movement of the hind motor, is not sufficient to lift the feet up.

Altogether, the robot travels forward as expressed in a total positive displacement of front and hind feet in the horizontal direction during one cycle. The symmetry of the gait resembles that of a trot. The small velocity of this motion was chosen deliberately for practical reasons.
Figure 2.18: A) Schematic of the controller circuit; the circles stand for the basic elements shown in Fig. 2.2 They are coupled together in a ring, the output voltage of one element being the input-voltage of the next. B) shows the entire control circuit in detail. In this ring, two modes can be observed. C) and D) show the measured output voltages V1out to V4out for each mode. These outputs control the leg movements. Depending on the initial conditions, there can be one signal (of length T1, marked with *) propagating through the ring, as shown in C), or two signals (of lengths T1 and T2, marked with ° and * respectively) as shown in D). When the outputs are connected to the motors as shown in A), two different gaits are obtained: a slow walk in case C) and a faster trot in case D) as is shown in Fig. 2.19 and Fig. 2.20, respectively.
Figure 2.19: Data of leg trajectories and internal voltages of the walking robot (trot-like gait). Two cycles are enlarged for clarity. For details see text.
Figure 2.20: Data of leg trajectories and internal voltages of the walking robot. One cycle is enlarged for clarity. For details see text.
Figure 2.20 shows the data for the second possible mode of the controller. In this case the robot moves much slower and backwards.

The voltage traces show that the left front foot moves forward, when the output of ‘element 4’ of the ring (V4out) is low, because this causes the front motor to turn the left leg forward. Also the hind leg moves backwards due to slip. During the next time interval the signal that propagates through the controller causes the next output-node to be low. This turns the front motor back, therefore moving the front foot backwards. During the following time interval the hind motor turns the hind left leg downwards, thereby lifting weight from the front left leg and enabling it to slip further backwards. In the last time interval the hind motor turns in the other direction, which does not effect the front left leg, but causes the hind left leg to move backwards. This movement results in a total negative displacement of the robot in the horizontal direction. The total velocity is significantly smaller than for the forward trot. This gait resembles a backward walk.

The gaits can be switched by changing the initial conditions of the oscillation. This can, for example, be done by a tactile sensor that briefly connects an input node of one of the inverters to V_{dd}. A possible use of this behavior is having the robot back away from obstacles it runs into. Other sensory information can modify the controllers behavior directly via changes in resistance, causing changes in the duration of the output signals and therefore changes in the coordination of the leg movements. These can, for example, lead to turning behavior. A tactile sensor monitoring one side of the robots body can therefore cause it turn away from sensed objects (or towards them, depending on the bias chosen by the designer) and likewise a set of photo-diodes, monitoring both sides of the robot, can result in a photo-tropic (or photo-phobic) behavior.

2.5.7 Discussion

Two controllers, each of which consists of four threshold elements were tested on a four-legged walking robot powered by two motors. Both controllers are very efficient, needing no software and only a small number of components to guarantee coordination between the leg movements leading to walking.

The controller which consists of a chain of two coupled oscillators is able to support a walk-like gait. The gait and also the direction of the robot can be changed directly with sensory input. The gait arises from a phase lag between front and hind girdle, which is set by the value of the coupling resistors. All walking gaits that have a phase lag of 180° between the legs within one girdle can be obtained by with this controller. These gaits are the walk, the trot and the gallop.

The controller that uses a ring of threshold elements can support two different
behaviors of the robot, a fast forward trot and a slow backward walk. The fact that the backward gait is slower than the forward gait resembles the way animals would move. Moving backwards, a creature needs lower speed and higher flexibility to turn. The slow backward walk offers this in comparison to the faster forward trot.

Some limitations that arise from mechanical problems have already been described, such as the lack of friction between feet and floor. Furthermore, it should be pointed out that the second controller (with ring architecture) has two significant disadvantages. Firstly, for smooth, stable walking one would like to be able to control the activity of the motors in such a way that an overlap of variable phase difference is possible, which can not be done with this controller. Secondly one would like to be able to model more than two gaits.

2.6 Conclusion

The oscillators that were analyzed and tested in this chapter can control walking robots. They offer the possibility to change the robots behavior using a variety of simple sensors. Our analysis revealed which architecture of a walking controller that is made from these oscillators is most suitable for gait control. A chain of four coupled oscillators can produce all the major gaits that four-legged animals have been observed to use. It is the minimal circuit required to control a rich variety of gaits, as controllers (of this kind) with fewer components can not achieve this goal.

Two controllers, one of ring and one of chain architecture, were tested on a walking robot. The machine had only two degrees of freedom (DOF) and thus two oscillators were sufficient to control it. The controller can be extended to one that drives a machine with one DOF per leg. This is the minimum number of DOFs, which enables us to emulate all the walking gaits of quadrupeds.

The parameters of the controllers which determine their output pattern were values of resistors in this chapter. Those can be changed either mechanically or electronically, with sensors. The way the sensors alter the behavior is fixed by the value of the resistances. This is useful, if one wants to change the behavior of the robot in a reflex like way. For more advanced usage of the walking machine and if one aims at implementing learning, it is desirable to make the controllers programmable electronically. Only then, an analogy between the spinal cord that receives input from the brain can be drawn to these controllers which could then receive input from another computing device. The next chapter introduces a chip which implements this idea using VLSI technology.
Chapter 3
Walking Gait Control Chip

Based on the circuits that were analyzed in Chapter 2, a chip that controls the walking behavior of a four-legged walking machine is constructed and tested. This chip is able to support different walking gaits of a four-legged walking machine. The chip is presented in Sec. 3.1. An analysis of the circuits on the chip (Sec. 3.3) provides an understanding of the chip's characteristics, which in turn enables us to investigate patterns that are useful with respect to locomotion, and that can be produced by the chip. The chip is designed such that it can be used in different configurations. The behavior of controllers with a chain architecture and of controllers with a ring architecture is investigated in Sec. 3.4.

3.1 The Chip

The Walking Gait Control (WGC) chip consists of two pairs of coupled oscillators, one single oscillator and two coupling circuits (see Fig. 3.1 to Fig. 3.3). These components can be connected in different ways by the user to yield different controllers. Some of the controllers which can be implemented this way are sketched in Fig. 3.6 and analyzed in Sec. 3.4.

The compactness of the circuits on the chip leaves a large area, which is used for on-chip capacitors. However, it is not necessary to make the capacitors this big. Thus, the capacitors are not a limiting factor for the implementation of larger networks of oscillators on one chip.

3.2 Data Acquisition

The output of the chip is measured by connecting it to a data acquisition card (Lab-PC+ board, available from National Instruments) in a personal computer (PC). A custom routine, written in C, calculates the frequency and the duty cycle of each output trace as well as the phase lags between output traces.
Figure 3.1: WGC chip layout. The oscillators are located in the upper left corner of the lower right quadrant. A blow up of this region is shown in Fig. 3.2. Most of the space is covered by large capacitors. The chip was fabricated in a CMOS 1.2 \( \mu \) process. The dimensions of the chip are 2mm x 2mm.
Figure 3.2: A blow up of the region of the WGC chip layout (Fig. 3.1) in which the oscillator circuits are situated. The oscillator circuits (compare Fig. 3.4) are located in the first, third and fifth row from the top. In the rows between them are the coupling circuits (compare Fig. 3.5). This layout is extremely compact, containing less than 80 transistors.
Figure 3.3: Architecture of the WGC chip. Each of the dotted lines is connected to a pad. The round symbols stand for oscillators (circuit diagram in Fig. 3.4) and the arrows stand for the circuits which couple two oscillators (circuit diagram in Fig. 3.5).
3.3 Circuit Analysis

The control parameters of the chip are two distinct bias voltages per oscillator and two more per coupling circuit (Fig. 3.3). This section elaborates how these voltages set the frequency and the duty cycle of a single oscillator as well as the phase shifts between oscillators. Specifically, these voltages determine the patterns which can be utilized for legged locomotion control. The frequency determines the frequency of the leg movement, the walking gait of the robot is defined by the phase shifts between the oscillators and the duty cycles of the oscillators control the direction of motion of the robot. The frequency of an oscillator depends exponentially on the oscillator's bias voltages, while the duty cycle depends exponentially on the difference between the oscillator's two bias voltages. The phase lag between two
Figure 3.5: Two oscillators are coupled with a transmission gate. This coupling affects only oscillator \( j \) (see Sec. 3.3.3). The gate voltage on the n-fet of each transmission gate is set to be the complementary voltage of the p-fet of the same transmission gate. This is done by the circuits which are drawn next to the transmission gates. These circuits are controlled by the bias voltages \( V_{b,ij,l} \) and \( V_{b,ij,r} \), respectively. They copy the voltages \( V_{b,ij,l} \) and \( V_{b,ij,r} \) to nodes 2 and 4, respectively, while the voltages at nodes 1 and 3 are \((V_{dd} - V_{b,ij,l})\) and \((V_{dd} - V_{b,ij,r})\), respectively.
Figure 3.6: Two different kinds of control architectures which are amongst those that can be implemented with the Walking Gait Control chip. (A) ring of coupled oscillators and (B) chain of coupled oscillators. Their behavior is analyzed in Sec. 3.4

oscillators depends on the bias voltages of the oscillators as well as the bias voltages of the coupling circuit. The function is not strictly exponential, but dominated by an exponential term.

3.3.1 Notation

A notation is adopted in this chapter that follows the same logic as the notation of the previous chapter. As before, the voltages at the input nodes to the inverters are called $V$, those at the nodes on the other side of the capacitors ($C$) are called $V_{in}$ and those at the inverter outputs are called $V_{out}$. The gate voltage of the n-fet transistors between node $V$ and ground (those transistors that are not part of the inverter) is called $V_b$.

The variables on the two sides of an oscillator are distinguished by the subscripts $r$ and $l$ while the variables of two different oscillators are distinguished by subscripts $i$ and $j$, as before. These subscripts are omitted in equations which address only one side of the oscillator or only one oscillator, respectively.

Two oscillators, $i$ and $j$, are coupled with transmission gates that connect $V_{out,i}$ and $V_j$. The bias voltage that controls the current through the transmission gate is called $V_{b,ij}$.
3.3.2 The Oscillator

The oscillator is very similar to the one discussed in Chapter 2, the main difference being that the resistor to ground is replaced by the n-fet transistor to ground (M1 of Fig. 3.4).

We want to calculate the period of the oscillator as a function of the gate voltage on M1, $V_b$. First, note that $V_b < 0.8V$ in the frequency range that is used for locomotion (ca. 0.5 Hz to 2 Hz). Hence, we can assume that the transistor M1 operates in the sub-threshold regime. We are interested in computing the time, $T_k$, (where $k \in \{r, l\}$ indicates the side of the oscillator) that it takes $V_k$ to decay from its peak voltage $V_{max}$ at $t = 0$ to the threshold voltage of the inverter, $V_{th}$. We assume that $t = 0$ is just after the jump from ground to the supply voltage, $V_{dd}$, occurred at node $V_{k,in}$. Thus, during the time $0 \leq t \leq T_k$, the voltage at node $V_{k,in}$ is constant and equal to $V_{dd}$. Moreover, $V_{th} \leq V_k \leq V_{max}$. $V_{th}$ can be calculated from the process parameters [124]

$$V_{th} = \frac{V_{dd} + V_{tp} + V_{tn} \sqrt{\beta_n/\beta_p}}{1 + \sqrt{\beta_n/\beta_p}}$$

where $V_{tp}$ and $V_{tn}$ are the threshold voltages for the inverter’s p-fet (M2) and n-fet (M3), respectively, while

$$\beta_i = \mu_i C_{ox} \frac{W_i}{L_i}, \quad i \in \{n, p\}$$

with $\mu_i$, the mobility of the charge carriers, $C_{ox}$, the gate oxide capacitance, $W_i$, the width and $L_i$, the length of the transistors. Both p- and n-fet have the same geometry in the present layout ($W_n = W_p$ and $L_n = L_p$). The threshold voltage of the inverter becomes

$$V_{th} = \frac{V_{dd} + V_{tp} + V_{tn} \sqrt{\mu_n/\mu_p}}{1 + \sqrt{\mu_n/\mu_p}} = 1.345V$$

M1 operates in saturation [79], because $V_k > 4kT/q$ during $0 \leq t \leq T_k$. The drain current through transistor M1 is given by [79].

$$I_T = I_0 e^{\frac{Q}{kT}V_{th}}$$

(3.1)

where $T$ is the temperature, $k$ the Boltzmann constant and $q$ the charge. Applying Kichhoff’s current law at node $V_k$, the differential equation for $V_k$ becomes:

$$\dot{V}_k = -\frac{I_0}{C} e^{\frac{Q}{kT}V_{th,k}}$$

$$V_k(t = 0) = V_{max}$$

$$k \in \{l, r\}$$
Figure 3.7: Period of the oscillation (asterix = data) as a function of the bias voltage on the n-fet transistor between node $V$ and ground. For simplicity the bias voltages on both n-fets of the oscillator are set to the same value, $V_b$, which means that the duty cycle of the oscillation is 50%. The oscillation period follows eq. (3.5) (solid line). The capacitance of all the capacitors on the chip is $C = 515.987619 \times 10^{-12} F$, according to the process parameters. The values of $V_{\text{max}}$, $I_{\text{on}}$ and $\kappa$ are estimated with a least squares fit: $V_{\text{max}} = 3.23 \, V$, $I_{\text{on}} = 2.2095 \times 10^{-16} A$ and $\kappa = 0.6202$. 
This equation is solved by

\[ V_k(t) = V_{\text{max}} - \frac{I_0}{C} e^{\frac{q}{kT} \kappa V_{b,k}} t \]

With \( V_k(t = T_k) = V_{\text{th}} \), we obtain \( T_k \):

\[ T_k = (V_{\text{max}} - V_{\text{th}}) \frac{C}{I_0} e^{-\frac{q}{kT} \kappa V_{b,k}} \quad (3.2) \]

The period of the oscillator is given by the sum of the signal times on both sides (subscripts \( l \) and \( r \) for left and right side) of the oscillator circuit

\[ P = T_r + T_l = \frac{C}{I_0} (V_{\text{max}} - V_{\text{th}}) (e^{-\frac{q}{kT} \kappa V_{b,l}} + e^{-\frac{q}{kT} \kappa V_{b,r}}) \quad (3.3) \]

This equation assumes that \( V_{\text{max}} \) and \( V_{\text{th}} \) are the same for both inverters of the oscillator. The inverters have the same geometry, thus this assumption is reasonable. With the same assumption, the negative duty cycle with respect to \( T_r \), as defined in Chapter 2, is given by

\[ D_r = \frac{T_r}{P} = \frac{1}{1 + e^{\frac{q}{kT} \kappa (V_{b,r} - V_{b,l})}} \quad (3.4) \]

and the same for \( D_l \), with \( r \) and \( l \) exchanged. Note that if all components of the oscillator have the same value and the bias voltages are the same on both sides \( (V_{b,l} = V_{b,r} = V_b) \), then the duty cycle is \( \frac{1}{2} \) (or 50 \%) and the period becomes (compare Fig. 3.7)

\[ P = \frac{2C}{I_0} (V_{\text{max}} - V_{\text{th}}) e^{-\frac{q}{kT} \kappa V_b} \quad (3.5) \]

i.e. by changing the bias voltage \( V_b \), the oscillation period is changed exponentially.

### 3.3.3 The Coupling

Two oscillators are coupled with a transmission gate between the output nodes of one oscillator \( i \) and the input nodes to the inverters of the other oscillator \( j \) (see Fig. 3.5), resembling the way the oscillators were coupled with resistors in Chapter 2. The bias voltage of the p-fet of the transmission gate is \( V_{b,ij} \). The bias of the n-fet of the transmission gate is set to \( (V_{dd} - V_{b,ij}) \) by an additional circuit (see Fig. 3.5). This circuit works in the following way: The input voltage of this circuit is voltage \( V_{b,ij} \). (Since there are two such circuits, one for each transmission gate, the input voltages are labeled with additional subscripts \( r \) and \( l \) in Fig. 3.5.) We can assume that the transistors are in saturation, because \( V_{b,ij} > 4kT/q \). Furthermore, let us assume that the transistor parameters, \( \mu, \kappa, w/l, I_o \) are constant and identical.
for each transistor that is of the same type (n or p). Then the currents through the transistors of this circuit are

\[ I_1 = I_{0p} e^{\frac{q}{kT} \kappa (V_{dd} - V_{b,ij})} \]
\[ I_2 = I_{0n} e^{\frac{q}{kT} \kappa V_n} \]
\[ I_3 = I_{0n} e^{\frac{q}{kT} \kappa V_n} \]
\[ I_4 = I_{0p} e^{\frac{q}{kT} \kappa (V_{dd} - V_p)} \]

where \( V_n \) and \( V_p \) stand for the gate voltages of the n-fet and the p-fet of the transmission gate, respectively. For the circuit on the left side in Fig. 3.5, \( V_n \) is the voltage at node 1 and \( V_p \) is the voltage at node 2. For the circuit on the right side in Fig. 3.5, \( V_n \) is the voltage at node 3 and \( V_p \) is the voltage at node 4. The currents \( I_1 \) and \( I_2 \) must be identical and similarly, \( I_3 \) and \( I_4 \) must be identical. Furthermore, the currents \( I_1 + I_2 \) and \( I_3 + I_3 \) must be equal. Thus, it follows that \( I_1 = I_4 \). Using this, we see that

\[ I_1 = I_2 \rightarrow V_n = V_{dd} - V_{b,ij} \]
\[ I_1 = I_4 \rightarrow V_p = V_{b,ij} \]

At the low frequencies which are relevant for locomotion, the coupling only affects oscillator \( j \) significantly, because the inverters of oscillator \( i \) act as impedance buffers and therefore the voltage on node \( V_{out,i} \) is not significantly altered by connecting the transmission gate. For ease of reference let us call oscillator \( i \) the master oscillator and oscillator \( j \) the slave oscillator, like in Chapter 2.

Phase lag between two oscillators

The phase lags, \( \phi_r \) and \( \phi_l \), between two oscillators (previously defined in Chapter 2, (2.18)) are given by

\[ \phi_r = \frac{T_{D,r} + T_{i,r}}{P} = \frac{T_{D,r}}{P} + D_{i,r} \]  
(3.6)

and the same for \( \phi_l \) with \( r \) and \( l \) exchanged. To calculate \( T_D \), consider that part of the oscillation during which the voltage at node \( V_{j,in} \) is at ground (0V). Let the time \( t = 0 \) be right after \( V_{out,i} \) switched from ground to \( V_{dd} \). When this step occurs, current flows into node \( V_j \) through the transmission gate. The voltage \( V_j \) increases and reaches the threshold voltage of the inverter, \( V_{th} \) after a time \( T_D \). For that range of the bias voltage on the transmission gate which we are interested in, the frequency of the slave oscillator is the same as the frequency of the master oscillator. Then the shift in time between the two oscillations, \( T_D \), results in a phase lag as defined in eq. (3.6).
Figure 3.8: A: The voltage output of inverter \( i \) \( (V_{out,i}) \) is the input to the transmission gate that couples inverter \( i \) to inverter \( j \). On the other side of the transmission gate is node \( V_j \). When the voltage \( V_{out,i} \) steps from 5V to 0V, the voltage \( V_j \) falls due to charge flow through the transmission gate. \( V_j \) is the input to one of the inverters of oscillator \( j \). When the voltage \( V_j \) reaches the inverter threshold, it causes the inverter to switch from low to high. B: Sketch of the output traces of two coupled oscillators. Signal times (e.g. \( T_{i,l} \)) and delay times (e.g. \( T_{D,i} \)) are indicated.
Derivation of $T_D$

To be able to use the approximation that all transistors are in saturation, let us assume that $V_j = 0.1$ V (instead of 0 V) at time $t = 0$. This introduces a very negligible error in the phase lag we are going to determine as a function of $V_{b,ij}$. The assumption made is only correct if there was enough time for $V_j$ to discharge, between the last time $V_{in,j}$ switched from $V_{dd}$ to ground and time $t = 0$. This can be assumed to be the case for the range of frequencies (around 0.5 to 2 Hz) we are concerned about. The time it takes to fully discharge $V_j$ depends on $C$, $V_{b,j}$ and $V_{b,ij}$.

In summary, the assumptions about the state of the oscillator considered for the computation of $T_D$ are:

\[
\begin{align*}
V_{out,i}(t < 0) &= 0 \\
V_{out,i}(t > 0) &= V_{dd} \\
V_{in,j} &= 0
\end{align*}
\]

with the initial condition

\[
V_j(t = 0) = V_0 = 0.1 \text{ V}
\]

With these assumptions, the current through the transmission gate is given by

\[
I_{TG} = I_{on} e^{\frac{g}{kT} \kappa (V_{dd} - V_{b,ij})} e^{-\frac{q}{kT} V_j} + I_{0p} e^{\frac{g}{kT} \kappa (V_{dd} - V_{b,ij})}
\]

The current through the single transistor is the same as before (compare eq. (3.1)) and Kirchhoff's current law implies that we have to solve

\[
\dot{V}_j = D e^{-\frac{q}{kT} V_j} + E \quad ; \quad V_j(t = 0) = V_0
\]

with

\[
D = \frac{I_{on}}{C} e^{\frac{g}{kT} \kappa (V_{dd} - V_{b,ij})}
\]

\[
E = \frac{1}{C} (I_{0p} e^{\frac{g}{kT} \kappa (V_{dd} - V_{b,ij})} - I_{on} e^{\frac{g}{kT} \kappa V_{b,j}})
\]

This is done by

\[
V_j = \frac{kT}{q} \ln \left[ e^{\frac{q}{kT} Et} \left( e^{\frac{q}{kT} V_0} + \frac{D}{E} \right) - \frac{D}{E} \right]
\]

Recalling that $V_j(T_D) = V_{th}$ we obtain

\[
T_D = \frac{kT}{qE} \ln \left( \frac{D + E e^{\frac{q}{kT} V_{th}}}{D + E e^{\frac{q}{kT} V_0}} \right)
\]
To write $T_D$ as a function of the bias voltages, let us define

$$\alpha = \frac{kT}{q} \frac{C}{I_{on}}$$

(3.8)

$$\mu = \frac{I_{op}}{I_{on}} e^{\frac{\alpha}{kT} \kappa V_{dd}}$$

$$\rho(V) = (I_{on} + I_{op} e^{\frac{\alpha}{kT} V}) e^{\frac{\alpha}{kT} \kappa V_{dd}}$$

$$\sigma(V) = I_{on} e^{\frac{\alpha}{kT} V}$$

(3.9)

With these abbreviations, $T_D$ becomes

$$T_D = \frac{\alpha}{\mu e^{-\frac{\alpha}{kT} \kappa V_{b,ij}} - e^{\frac{\alpha}{kT} \kappa V_{b,j}}} \ln \left[ \frac{\rho(V_{th}) e^{-\frac{\alpha}{kT} \kappa V_{b,ij}} - \sigma(V_{th}) e^{\frac{\alpha}{kT} \kappa V_{b,j}}}{\rho(V_0) e^{-\frac{\alpha}{kT} \kappa V_{b,ij}} - \sigma(V_0) e^{\frac{\alpha}{kT} \kappa V_{b,j}}} \right]$$

(3.10)

$T_D$ is only defined, if the argument of the logarithm in (3.10) is greater than zero. This means that numerator and denominator are either both smaller or both greater than zero. This yields a constraint for the sum of $V_{b,j}$ and $V_{b,ij}$

$$V_{b,ij} + V_{b,j} < \frac{kT}{q \kappa} \ln \left[ (e^{-\frac{\alpha}{kT} V_{th}} + I_{op} e^{\frac{\alpha}{kT} \kappa V_{dd}}) \right]$$

OR

$$V_{b,ij} + V_{b,j} > \frac{kT}{q \kappa} \ln \left[ (e^{-\frac{\alpha}{kT} V_0} + I_{op} e^{\frac{\alpha}{kT} \kappa V_{dd}}) \right]$$

Note that with $V_0 = 0.1$ V and $V_{th} = 1.3453$ V, rough numerical values for the bounds are

$$V_{b,ij} + V_{b,j} \lesssim 4.75$$

OR

$$V_{b,ij} + V_{b,j} \gtrsim 4.86$$

This means that there is a small region of approximately 100mV in which $T_D$ diverges. The singularity can be observed experimentally, but it is not reached within the range of bias voltages applied for the use of the chip in this thesis. Since it is not possible to measure the values of $I_{op}$ and $I_{on}$ for every transistor on the WGC chip, these two parameters are estimated by data fitting.

**Behavior of $T_D$**

$T_D$ becomes zero when the transmission gate opens ($V_{b,ij} \rightarrow 0$). Then, the term in front of the logarithm in eq. (3.10) becomes $-\frac{\alpha}{\mu e^{\frac{\alpha}{kT} \kappa V_{b,ij}}}$. The denominator is
dominated by $\mu$, which is on the order of $10^{56}$ and thus the term vanishes which in
turn causes $T_D$ to go to zero.

Note that the phase shift between two oscillators $i$ and $j$ is different for two cases:
(i) in the circuit there is a master oscillator that has no transmission gate connected
to $V_t$ and therefore behaves according to eqs. (3.2) and (3.3). This is the case for
chains of coupled oscillators. Since we are interested only in oscillations where both
oscillators are entrained to the same frequency, eq. (3.2) substituted into eq. (3.6)
together with eq. (3.10) gives the phase shift for this case; (ii) all oscillators are
coupled to others via a transmission gate. As will become clear in Sec. 3.4.2, the
common period is different in this case.

3.3.4 Temperature dependency
The output of the chip, i.e. oscillation period, duty cycles and phase shifts are
temperature dependent. Engineering the chip such that its output becomes tem¬
perature independent is not necessarily a trivial task [28]. For some applications it
might be sufficient to mount the chip in a thermally isolated enclosure.

3.4 Patterns produced by rings and chains of os¬
cillators
We are interested in the patterns that those controllers which can be implemented
using the Walking Gait Control chip can produce. In particular, chain and ring
architectures are of interest. A controller with a chain architecture can support all
the major walking gaits that have been observed to be used by four-legged animals.
Those, together with the speed and direction of the robot are controlled by eight
parameters. A ring of four oscillators shows bistable behavior thus being able to
support two gaits.

3.4.1 Chains of oscillators
Chains of oscillators can produce phase lags between the oscillatory movement of
the legs of the walking machine in the range of 0.5 to 1. The phase lags are smooth
functions of the respective control parameters. Therefore transitions between dif¬
ferent gaits can either be smooth or abrupt, the later implemented by an abrupt
change of the control parameters.

Let $n$ oscillators be coupled to form a chain. Let the oscillators be numbered with
subscripts $i = 1, \ldots, n$. In the first oscillator there is no transmission gate connected
to node $V_1$. Hence, this oscillator has a period that is given by eq. (3.3).
Chains of symmetrical oscillators

A very simple requirement for the circuit is to produce a symmetric waveform for forward motion. This can be implemented by identical control parameters on both right and left side of the oscillators ($V_{b,rl} = V_{b,lr} =: V_{b,l}$ and $V_{b,rij} = V_{b,ijr} =: V_{b,ij}$). Then eq. (3.6) becomes

$$\phi = \frac{T_D}{P} + \frac{1}{2}$$

(compare Fig. 3.9 to Fig. 3.11). Substitution of $P$ (see (3.5)) and $T_D$ (compare (3.10)) into (3.11) yields

$$\phi = \frac{1}{2} + \frac{1}{2} \frac{kT}{q(V_{max} - V_{th})} \frac{e^{\frac{q}{kT} \kappa(V_{b,m} - V_{b,j})}}{\mu e^{\frac{q}{kT} \kappa(V_{b,ij} + V_{b,j})} - 1} \ln \left[ \frac{\rho(V_{th}) - \sigma(V_{th}) e^{\frac{q}{kT} \kappa(V_{b,ij} + V_{b})}}{\rho(V_0) - \sigma(V_0) e^{\frac{q}{kT} \kappa(V_{b,ij} + V_{b})}} \right]$$

Where the subscript $m$ denotes the master oscillator at the head of the chain. If the bias voltages for all oscillators are the same ($V_{b,i} = V_m \ \forall i \in \{1, \ldots, n\}$), $\phi$ depends only on the sum of $V_m$ and the bias voltage of the transmission gate

$$\phi = \frac{1}{2} + \frac{1}{2} \frac{kT}{q(V_{max} - V_{th})} \frac{1}{\mu e^{\frac{q}{kT} \kappa(V_{b,ij} + V_{b,j})} - 1} \ln \left[ \frac{\rho(V_{th}) - \sigma(V_{th}) e^{\frac{q}{kT} \kappa(V_{b,ij} + V_{b})}}{\rho(V_0) - \sigma(V_0) e^{\frac{q}{kT} \kappa(V_{b,ij} + V_{b})}} \right]$$

Walking Gaits controlled with a chain of four coupled oscillators

To generate different walking gaits as described in Chapter 1, the phase lags between single oscillators have to be adjusted to the appropriate values. Typical gaits like trot, walk, transverse and rotary gallop, canter, bounce and pronk can in principle be controlled by the output of the chip (Fig. 3.13). However, a problem arises here. To illustrate this, let us again consider symmetric oscillators, symmetric meaning that the parameter values on both right and left side of the circuit are the same. The duty cycle of a symmetric oscillator should be 50 %, because it is given by $D_i = T_i/(T_i + T_r) \ (i \in \{r, l\})$ and for symmetric oscillators, $T_l = T_r$. This is assuming an ideal chip with identical transistors.

In practice transistor mismatch is a problem in the development of aVLSI systems [79]. Transistors operated in the subthreshold regime show a large variability (as reported in [5]). The variability is attributed to several effects [88], not all of which can be eliminated by careful design [5].

The difference between the transistors of the chip causes the transmission gates on the two sides of the circuit to let a different amount of current pass when set to the same bias voltage. This results in different delay times $T_D$ on both sides of the circuit. Assuming that the symmetric master oscillator has a duty cycle of 50 %,
Figure 3.9: Phase lag (in units of 2 $\pi$) between the first two oscillators in a chain of 4 oscillators. The function given in eq. (3.11) (solid line) is compared to the data points. The parameters $V_{max}$ and $I_0$ are the ones determined in Fig. 3.7, $I_{op}$, as estimated by data fitting is $1.56 \times 10^{-19}$ A.
Figure 3.10: Phase lag (in units of $2\pi$) between oscillators 2 and 3 in a chain of 4 oscillators. The function given in eq. (3.12) (solid line) is compared to the data points. The parameters $V_{max}$ and $I_{on}$, used to calculate the master oscillator's period are the ones determined in Fig. 3.7. For the transistors of the slave oscillator, $I_{op}$ and $I_{on}$ are estimated by data fitting. $I_{op} = 6.5 \times 10^{-18} \text{ A}$ and $I_{on} = 10^{-17} \text{ A}$. 
Figure 3.11: Phase lag (in units of $2\pi$) between the last two oscillators in a chain of 4 oscillators. The function given in eq. (3.12) (solid line) is compared to the data points. The parameters $V_{\text{max}}$ and $I_{\text{on}}$, used to calculate the master oscillator’s period are the ones determined in Fig. 3.7. For the transistors of the slave oscillator, $I_{0p}$ and $I_{0n}$ are estimated by data fitting. $I_{0p} = 1.38 \times 10^{-19}$ A and $I_{0n} = 2 \times 10^{-17}$ A.
Figure 3.12: Duty cycle of oscillator two in a chain of 4 oscillators as a function of the phase shift (in units of $2\pi$) between oscillators one and two.

The slave oscillator must then have a duty cycle different from 50% (see Fig. 3.12). In practice however, the duty cycle of the symmetric master oscillator will also be slightly different from 50% when the transistors M1 on each side of each oscillator (see Fig. 3.4) are set to the same bias voltage, because they also let a different amount of current pass due to the same effect.

If the current through each transistor could be determined experimentally, the theoretical modeling could be improved to better describe the observed duty cycle. For practical reasons, the current through every transistor of the chip as a function of the bias voltage cannot be measured, because the transistors are not connected to bare pads. Therefore, another way has to be found for adjusting the parameters of the chip to produce exactly the desired pattern. In the configuration sketched in Fig. 3.13, the pace gait is trivial to produce. For this gait, the transmission gates must be open which can easily be assured by setting $V_{b,12} = V_{b,23} = V_{b,34} \leq 3.9$ V.

But the parameters for other gaits are harder to be determined.

The brute force method is to create a look up table by finely scanning the parameter space and measuring the resulting output traces of the chip with the automated

\footnote{If they were connected to bare pads, the chip would not operate, because the oscillation would break down due to a lack of an impedance buffering. Furthermore, the number of pads is limited to 40 in this package.}
Figure 3.13: Different walking gaits and the respective phase lags for a chain of oscillators (top left drawing). The numbers indicate the phase shift of the legs with respect to the front left leg according to Fig. 1.1.
Figure 3.14: The phase lag between oscillators 2 and 3 in a chain of 4 oscillators (grey scale) in the two dimensional parameter space spanned by the two bias voltages $V_{b_{ij}l}$ and $V_{b_{ij}r}$ of the transmission gates between oscillators 2 and 3. Areas in which the phase lag attains a value of roughly 0.6 (0.599 to 0.601) are surrounded by black lines in the graph. The resulting pattern produced by the controller corresponds to a transverse gallop, as shown in Fig. 3.15. The black spots (value 0) are artifacts. No reliable data is available at these points (see Sec. 3.2).
Figure 3.15: Output waveforms which control a movement similar to a transverse gallop, if connected to the motors of the walking machine as sketched in Fig. 3.13. The phase lags are: $\phi_1 = 0.5$, $\phi_2 = 0.6$ and $\phi_3 = 0.5$. The values of the bias voltages are obtained from Fig. 3.14.
data acquisition described in Sec. 3.2. This method of searching for the parameters which produce a desired pattern is very time consuming, because firstly the input parameter spaces are two dimensional which means that many different combinations have to be measured and secondly the data acquisition is noisy which means that each measurement has to be repeated many times to obtain an average with decent precision (see Sec. 3.2). However, it can be done. Let us take the transverse gallop gait as an example. To implement this gait, we need to adjust the phase lags between the oscillators to \( \phi_1 = \phi_3 = 0.5 \) and \( \phi_2 = 0.6 \) (see Fig. 3.13). As mentioned above, a phase lag of 0.5 is trivial to implement, such that we only have to search for those values for the parameters \( V_{b,23,l} \) and \( V_{b,23,r} \) which ensure that \( \phi_2 = 0.6 \). With an automated data acquisition, a fine scan of the two bias voltages took roughly one day (Fig. 3.14). The result shows that a transverse gallop can be controlled with the chip after the correct bias voltages were found with the automated scan (Fig. 3.15).

Due to its inefficiency the parameter search by fine scanning of the parameter space is certainly not the method of choice. Chapter 5 deals with finding better solutions to this problem.

### 3.4.2 Rings of oscillators

To investigate the behavior of rings of coupled oscillators implemented with the Walking Gait Control chip let us focus on a ring of four identical oscillators with identical parameters on both side of the circuit. We investigate how the frequency of the controller depends on the bias voltages of the transmission gates and we find that the controller can support 2 different gaits.

#### Rings of symmetric, identical oscillators

For reasons discussed in Sec. 3.4.1, let us investigate symmetric, identical oscillators. All the oscillators of the ring controller are coupled to other oscillators. They are all 'slave' oscillators. Therefore their oscillation period and with it the period of the whole controller is different from that of a 'master' oscillator.

#### Frequency of a slave oscillator

In order to compute the oscillation period of the slave oscillator, we calculate the time \( T_j \) it takes the voltage at node \( V_j \) to decay from its maximum value, \( V_{\text{max}} \) to the threshold voltage of the inverter, \( V_{\text{th}} \). \( V_{\text{max}} \) is obtained at node \( V \) when the outputs of the oscillator's inverters switch and hence a voltage step arises at node \( V_{\text{out},j} \). We can assume that, during this time, \( V_{\text{out},i} \) does not change. For some oscillators in the ring, \( V_{\text{out},i} = 0 \)V during this time and otherwise \( V_{\text{out},i} = V_{dd} \) during this time (see Figs. 3.17 and 3.18). Let us consider the case that \( V_{\text{out},i} = 0 \).
Figure 3.16: Oscillation period as a function of the bias voltage of the transmission gates. They are all set to the same value, $V_{bias}$. The bias voltages of the transistors M1 on each side of each oscillator are all set to $V_0 = 0.2$ V
Using the same approximation as above (all transistors operate in sub-threshold and saturation), the current through the transmission gate is given by

$$I_{TG} = I_{on} e^{\frac{q}{kT} (V_{dd} - V_{b,ij})} + I_{op} e^{\frac{q}{kT} V_{b,ij}} - (1 - \kappa) V_{dd} e^{\frac{q}{kT} V_j}$$

The current through the single transistor to ground remains the same as in (3.1). Applying Kirchhoff’s current law we obtain

$$\dot{V}_j = -\frac{1}{C} (I_{TG} + I_T)$$

with the initial condition

$$V_j(t = 0) = V_{max}$$

Defining

$$A = \frac{I_{op}}{C} e^{\frac{q}{kT} V_{b,ij}} - (1 - \kappa) \frac{q}{kT} V_{dd}$$

$$B = \frac{I_{on}}{C} (e^{\frac{q}{kT} (V_{dd} - V_{b,ij})} + e^{\frac{q}{kT} \kappa V_{b,j}})$$
Figure 3.18: Output waveforms of a controller with ring architecture, mode 2. $V_b = 0.4V$, $V_{bias} = 4.3V$
Eq. (3.13) can be written as

\[ \dot{V}_j = -A e^{\frac{q}{kT} V_j} - B \quad ; \quad V_j(t = 0) = V_{\text{max}} \]

which is solved by

\[ V_j(t) = -\frac{kT}{q} \ln \left[ e^{\frac{q}{kT} B t} \left( e^{-\frac{q}{kT} V_{\text{max}}} + \frac{A}{B} \right) - \frac{A}{B} \right] \]

Remembering \( V_j(t = T_j) = V_{\text{th}} \) yields

\[ T_j = \frac{kT}{qB} \ln \frac{A + B e^{-\frac{q}{kT} V_{\text{th}}}}{A + B e^{-\frac{q}{kT} V_{\text{max}}}} \]

To show that \( T_j \) depends on the bias voltages let us define

\[ \beta = e^{\frac{q}{kT} \kappa V_{dd}} \]

\[ \gamma(V) = I_{0p} e^{\frac{q}{kT}(\kappa - 1)V_{dd}} + I_{0m} e^{\frac{q}{kT} \kappa V_{dd} - \frac{q}{kT} V} \]

(3.14)

With these abbreviations the signal time becomes, using (3.8) and (3.9)

\[ T_j = \frac{\alpha}{\beta e^{-\frac{q}{kT} \kappa V_{b,ij} + \frac{q}{kT} \kappa V_{b,jk}} \ln \frac{\gamma(V_{\text{th}}) e^{-\frac{q}{kT} \kappa V_{b,ij} + \sigma(-V_{\text{th}}) e^{\frac{q}{kT} \kappa V_{b,jk}}}}{\gamma(V_{\text{max}}) e^{-\frac{q}{kT} \kappa V_{b,ij} + \sigma(-V_{\text{max}}) e^{\frac{q}{kT} \kappa V_{b,jk}}}} \]  

(3.15)

If the bias voltages of the transistors M1 on both sides of the oscillator circuit (see Fig. 3.4) and the bias voltages of the transmission gates on both sides of the circuit are identical, then the period is \( P = 2T \). Otherwise, the period is the sum of the signal times of each half of the circuit, \( P = T_{j,1} + T_{j,2} \). The negative duty cycle with respect to \( T_{j,k} \) (\( k \in \{r, l\} \)) is \( D_{j,k} = T_{j,k}/(T_{j,1} + T_{j,2}) \).

Behavior of ring controllers

For symmetric oscillators, the period is given by \( P = 2T \), where \( T \) follows eq. (3.15). The dependency of \( T_j \) on the control voltages \( V_{b,ij} \) and \( V_{b,j} \) is dominated by the denominator of the factor \( 1/(\beta e^{-\frac{q}{kT} \kappa V_{b,ij}} + e^{\frac{q}{kT} \kappa V_{b,j}}) \), because the logarithmic term is almost constant. For the small values of \( V_{b,j} \) that result in oscillation frequencies adequate for locomotion (\( V_{b,j} \) roughly around 0.35 V and 0.4 V), the term \( \beta e^{-\frac{q}{kT} \kappa V_{b,ij}} \) dominates the denominator in the expression above. Hence the frequency of the controller depends roughly exponentially on the bias voltages of the transmission gates, \( V_{b,ij} \) (Fig. 3.16) and is not significantly affected by changes in \( V_{b,j} \). At those frequencies interesting for robotic locomotion (ca. 0.5 Hz - 2 Hz), the controller is bistable for most combinations of the parameters \( V_{b,ij} \) and \( V_{b,j} \), sup-
Figure 3.19: Circuitry required to implement the sensory feedback loop, consisting of two comparators and two n-fet MOS transistors. If $V_{out,l} = 5V$ and $V_{out,r} = 0V$, then the voltage of the potentiometer, $V_{pot}$, increases. If $V_{pot} < V_{max}$, then the gate voltage on transistor M1 is 0V, and M1 is turned off. If $V_{pot} > V_{max}$, then the gate voltage on transistor M1 is 5V, and M1 is turned on. This causes $V_{out,l}$ to go to 0V, and stops the leg from moving, as the inputs to both sides of the motor are 0V. Similarly, when $V_{out,l} = 0V$ and $V_{out,r} = 5V$, the voltage $V_{pot}$ decreases. In this case, as $V_{pot}$ decreases below $V_{min}$, $V_{out,r}$ is shorted to 0V, and the leg stops from moving. Thus the voltages $V_{min}$ and $V_{max}$ set the range for $V_{pot}$ in which the leg can move. They are 'voltage boundaries'.

porting two different modes one of which (mode 1) resembles a walk gait (Fig. 3.17). The second mode (mode 2) occurs with a much higher frequency (Fig. 3.18). If one was to drive a robot using these two different oscillation modes, then connecting $V_{out,l}$ to the motor of the leg it is supposed to move with opposite polarity than the other three waveforms are connected to their legs, would yield a fast trot gait for mode 2 and a slow walk gait for mode 1.

Compared to the wide range of gaits that a chain of 4 oscillators can control, the performance of a ring is rather limited.

3.5 Sensory feedback

Experiments with decerebrate cats have shown that the timing of the initiation of the swing phase (and of the termination of the stance phase) is influenced by sensory feedback. There are two hypotheses regarding the mechanism and the receptors involved in it [54]. One hypothesis states that the swing phase only begins when extensor muscles are being unloaded at the end of the stance phase (see [24], [48], [31], [91] and [90]), which means that force sensitive receptors are playing the crucial role. The other hypothesis sees hip extension as a critical signal for the initiation of the swing phase (see [4], [46], [54] and [68]).

The robot on which the chip is tested (see Chapter 4) contains potentiometers which measure the angular displacement of the leg. They give information about
Figure 3.20: Leg movement without feedback. Upper trace: control signal; voltage at one side of the motor, the voltage on the other side is the mirror image of this step function. Lower trace: voltage measured at the potentiometer attached to the leg. This voltage indicates that the leg moves in one direction, when the control signal is high, until it hits the mechanical boundary.
Figure 3.21: Leg movement with feedback. Upper trace: control signal; voltage at one side of the motor, the voltage on the other side is the mirror image of this step function. Lower trace: voltage measured at the potentiometer attached to the leg. With feedback, the leg also moves in one direction when the control signal is high, but it stops if the output of the potentiometer, $V_{pot}$, increases beyond $V_{max} = 1.565V$. The voltage boundaries $V_{max}$ and $V_{min}$ (1.225 V) are chosen such that the leg stops moving shortly before it touches the mechanical boundary.
the location of the leg relatively to the body. This signal can be interpreted as giving information similar to that about the hip extension in the biological system. Using these simple sensors, we can build a direct sensory feedback loop, incorporating the output of the sensor into the circuitry that drives the leg. This yields a system with a different behavior compared to the open loop system. The feedback can be used to avoid contact with the mechanical stops which constrain the maximal excursion of the leg. The additional circuitry required for this feedback loop is straightforward (see Fig. 3.19). Without feedback, the leg bumps into the mechanical boundaries (see Fig. 3.20), but with feedback this is avoided and the amplitude of the leg movement is regulated by the voltage boundaries $V_{\text{min}}$ and $V_{\text{max}}$ (see Fig. 3.21).

Using sensory information on hip position in a feedback loop as described demonstrates how a system composed of the WGC chip and a robot allows for inclusion of feedback from simple sensors. More generally, information provided by the sensors could also be used for more intricate control tasks. Recently, Lewis et al. [72] proposed an adaptive motorneuron stage which is located between the (square wave) output of an oscillator and the motor of a leg, receiving input from a LVDT sensor, which monitors the position of the hip. This circuit mimics the spiking outputs of two motor neurons, the frequency of which control the forward and the backward motor speed, respectively. This module could be used in combination with the WGC chip, replacing the circuit shown in Fig. 3.19. The advantage of the circuit presented here (see Fig. 3.19) lies in its simplicity and in the fact that it is made from cheap 'off-the-shelf' components. On the other hand, the circuit presented in [72] allows for adaptive control of speed, a feature which is not addressed by the circuits presented here. Hence, interfacing the WGC chip with the motorneuron circuit of [72] would be a sensible extension of the present work.

3.6 Conclusion

The Walking Gait Control chip which was presented and analyzed in this chapter can control a four-legged robot and support all the major walking gaits that four-legged animals have been observed to use. The chip takes constant analog voltages (= control parameters) as input and produces square wave voltages as output. Frequency and duty cycle of the oscillations as well as phase relationships between the oscillations depend on the input and determine the robot’s walking behavior.

The WGC chip’s input can be provided by:

- Analog chips (e.g. neuromorphic sensor chips, see Chapter 1.2.2)
- Sensors the output of which can be transformed into an analog voltage
- Digital processors (via digital to analog converters (DACs)).
Thus the chip can be used on a legged robot in combination with a variety of other devices. Sensory feedback can be incorporated into the controller either directly by using a voltage signal from the sensors to modify the chip’s input voltages, or indirectly by measuring the sensory signal with a PC-card, processing the sensory data on a digital processor and setting the chip’s input voltages via DACs.

The fact that digital machines can be interfaced to the WGC chip is also intended to be a design tool for the further development of the whole, autonomous walking robot. To develop a new algorithm for any task that the robot must perform, it is often faster and cheaper to implement the algorithm using digital hardware than to implement it on a dedicated chip, because of the chip fabrication time and costs. This strategy is particularly appropriate if an algorithm is modified during the development phase or if several algorithms are tested and compared. Thus, I propose a development strategy for a fully analog (neuromorphic) walking machine, using the WGC chip interface: First, find a suitable algorithm through implementation and testing on a digital processor. After the algorithm is found, it can be implemented as a neuromorphic chip, communicating with the WGC chip and also with either the sensors or the digital processor. In this way, a fully analog, autonomous machine can be developed which will be able to solve more and more complex tasks.

The circuit analysis of the WGC chip enables one to determine approximately the values of the control parameters needed for a particular behavior. They can not be determined precisely due to a lack knowledge of the exact values of the dark currents $I_0$ of each transistor on the chip. The parameter space in the vicinity of the approximate values of the control parameters can be scanned and the resulting waveform can be measured and compared to the desired one using an automatic data acquisition system. The utility of this method was demonstrated. However, this procedure is rather time consuming and not very elegant. The remaining challenge therefore is to enable the robot to find those parameters that lead to a desired behavior through some mechanism of adaptation or learning. These issues are addressed in Chapter 5.

Another question that needs to be addressed is that of the particular mechanical features of the robot who’s leg movements are controlled by the Walking Gait Control chip. Despite the chip being able to control a wide range of walking gaits, it might be the case that due to the mechanical properties of the robot not all of them result in efficient locomotion. Which ones do, depends on the particular mechanics of the machine. The chip is implemented on a four-legged robot with one motor per leg. Chapter 4 investigates which movements the robot executes given the output of the WGC chip.
Chapter 4

Four-Legged Walking Machine

A four-legged walking robot is constructed (see Sec. 4.1) in order to provide a test platform for the Walking Gait Control (WGC) chip (see Chapter 3). The mechanics of this robot are modeled (see Sec. 4.3) to provide an understanding of how the output voltages from the WGC chip move the robot in space. The control signals coordinate the legs of the robot for different gaits (see Sec. 4.2).

4.1 The Robot

The robot's body is an aluminum frame of variable height, with a length of 12 cm and a width of 6.2 cm. The frame holds four DC motors which drive the legs of the machine (see Fig. 4.1). The motors are standard servo motors that were modified such that their input signal specifications are compatible with the output of the WGC chip. Buffer circuits are used to decouple chip and motor signals and to supply enough power to the motors. An aluminum leg is attached to each motor, at right angles to the plane of the body, ending in a foot that is 2 cm high, 2 cm wide and 1.5 cm long. Leg and foot together have a length of 6.5 cm, measured from the motor axis to the bottom of the foot. Each foot has a small electromagnet which is activated during the stance phase of the leg and deactivated during the swing phase. The robot walks on a metal floor such that the electromagnet increases the friction during the stance phase\(^1\). This is a way of breaking the symmetry of the force the machine exerts on the floor to make it move in one distinct direction, given that it has only one degree of freedom per leg. A similar effect can be obtained using asymmetric, elastic legs (see [118]). The magnetic method was chosen, because it simplifies the mechanical construction and analysis while at the same time making it possible to move the robot forward efficiently. The electromagnet is an additional, but passive, degree of freedom per leg. It is called a 'passive' degree of freedom, because it is phase coupled to the control signal. The magnet is always turned on

\(^1\)Stance phase = Leg moves in the opposite direction of the intended locomotion and propels the body forward. Swing phase = Leg moves back into the position from which it can start the stance phase. In animals the foot is typically lifted off the ground during this phase.
Figure 4.1: Picture of the walking machine.
during the stance phase and always turned off during the swing phase. A biological motivation for this additional, passive degree of freedom can be found if one looks at the foot of the stick insect ([26]). The foot of the stick insect is composed of several cylinder-like compartments that fit into each other. A tendon runs through the foot and is attached beneath the joints which connect those compartments. When the leg moves backwards, the tendon pulls and causes the foot to contract. The ends of the compartments of the foot then push into the ground like small hooks, and there is an additional large hook at the tip of the foot. This increases the friction between foot and ground so much that it enables the stick insect to walk up walls.

4.2 Gait Control using the WGC chip

The walking gaits of this robot which has one degree of freedom per leg are defined by the phase relationships between the leg displacements under load conditions\(^2\). Potentiometers are attached to each motor. The voltage measured at a potentiometer is proportional to the angular displacement of the corresponding leg. The phase relationships of the four voltage traces obtained from the potentiometers characterize the gait. This data, together with data from the chip controlling the inter-leg coordination necessary for all the respective gaits (as shown in Fig. 1.1) confirms that each of the gaits which can be controlled by the output of the WGC chip (as discussed in Sec. 3.4.1) is actually expressed in the movement of the robot (see Fig. 4.2 to Fig. 4.9). In each figure, the four traces of the leg movements are overlaid to aid the discrimination of the phase relationships between them. Note that raw data is plotted, which is subject to noise caused by the data acquisition (see Sec. 3.2). The numbers given for the period, \(P\) and the phase shifts, \(\phi_i\) (\(i = 1, \ldots, 3\)), of the oscillators on the chip are obtained by averaging over 20 periods. Thus the data displayed can deviate slightly from these numbers. For each gait, we consider forward motion, implying that the legs' duty cycles are roughly 50%. Gait stability is demonstrated for the trot gait under no-load condition (see Fig. 4.10) and load condition (see Fig. 4.11). The other gaits are equally stable.

\(^2\)Load condition = Robot walks on the floor. No-load condition: Robot's body is supported, legs swing free in the air.
Figure 4.2: Trot, $P = 1.1s$. For each of the four legs, the control voltages (upper plot) and the voltage of the potentiometers which are attached to the motors are plotted as a function of time. The voltage of the potentiometers is normalized to the interval $[0,1]$. It is proportional to the angle $\varphi(t)$, which ranges from 0 to $\varphi_{\text{max}}$ and increase in angle means that the leg moves backwards with respect to the body's direction of movement, thus propelling the body forward (stance phase). The four legs are distinguished by the abbreviations: left front (LF; dots), right front (RF; circles), left hind (LH; crosses) and right hind (RH; stars). Legs LF and RH move in phase with each other, legs RF and LH move in phase with each other and legs LF and RF move 180° out of phase.
Figure 4.3: Forward walk, $P = 0.89s$. Plot legend as in Fig. 4.2. The legs move with a phase shift of roughly 90° in succession: LF, RH, RF, and finally LH.
Figure 4.4: Forward Canter. $P = 0.58s$. Plot legend as in Fig. 4.2. Legs LF and RH move in synchrony. Leg RF follows with a phase shift of roughly 0.3 (with respect to LF) and LH follows with a phase shift of roughly 0.7 (with respect to LF).
Figure 4.5: Forward Pace. $P = 0.85s$. Plot legend as in Fig. 4.2. Leg pairs on the same side of the body move in synchrony. There is a phase shift of roughly 0.5 between the two leg pairs.
Figure 4.6: Forward Transverse Gallop. $P = 0.63s$. Plot legend as in Fig. 4.2. The legs move in succession LF, RF, LH, RH with phase lags of roughly 0.1 between LF and RF as well as between LH and RH, and 0.5 between LF and LH.
Figure 4.7: Forward Rotary Gallop. $P = 0.65s$. Plot legend as in Fig. 4.2. The legs move in succession LF, RF, RH, LH with phase lags of roughly 0.1 between LF and RF as well as between RH and LH, and 0.5 between LF and RH.
Figure 4.8: Forward Bound. $P = 0.93s$. Plot legend as in Fig. 4.2. Both legs of the same girdle move in synchrony. There is a phase shift of roughly 0.5 between the two leg pairs.
Figure 4.9: Pronk. $P = 0.86$s. Plot legend as in Fig. 4.2. All four legs move in synchrony.
Figure 4.10: Plot of the angular displacement of left front vs. angular displacement of right front leg under no-load condition. The displacement is measured with potentiometers; the measured voltage is normalized to one. The trajectories of both legs are similar for each cycle. They lie on a noisy attractor, the structure of which is expressed in this measurement which contains 2778 data points.
4.3 Kinematic Model of the Robot

The output voltages of the WGC chip which drive the robot’s leg motors are known, as well as the angular movements of the legs which are measured with potentiometers attached to the leg motors. The motion of the robot’s body is modeled based on this knowledge.

Simplification 1: The leg movements of the robot move the robot’s body both horizontally and vertically. The vertical movement is small. Hence it will be neglected in the model. This means that we only model movements of the robot’s body in the plane. A projection of the robot’s body onto the horizontal plane (x-y plane) yields a rectangular shape of length $l$ and width $w$ (see Fig. 4.12 C). Assume that the legs are attached at the corners of this rectangle and call the corners the shoulders, $S_i$, $i = 1, .., 4$. The movement of the robot’s body is parameterized by the two dimensional vector, $\vec{c}$, that denotes the center of gravity and the angle, $\phi$, between the symmetry axis going through the robot’s body from rear to front and the x-axis (see Fig. 4.12).

Simplification 2: The robot’s legs swing beneath its body, executing a one dimensional movement on a circle with radius $a$. The movement of each leg is parameterized by the angle $\varphi_i(t)$, $i = 1, .., 4$. The maximal angular displacement $\varphi_{max}$ is the
Figure 4.12: Sketch of the model of the robot. A: World coordinates \((x,y,z)\), body centered coordinates, \((x',y',z')\), with origin at the center of gravity \((\vec{c})\) of the robot's body. Feet \((\vec{f}_i)\) and shoulders \((\vec{S}_i)\), \(i = 1,...,4\). B: Projection onto the \((x,z)\) plane. Length \((l)\) and height \((h)\) of the robot's body. \(a\): length of the legs. The angle \(\varphi_1\) describes the excursion of leg 1. C: Projection onto the \((x,y)\) plane. Width \((w)\) and length \((l)\) of the robot's body. The angle \(\phi\) is the angle between \(x\) axis and \(x'\) axis.
same for each leg and is set by the position of mechanical stops. Measurement of \( \varphi_i(t) \) reveals that \( \varphi_i(t) \) can be approximated with a piecewise linear function (see Fig. 4.13). Let us consider one single leg and let its movement start at time \( t_0 \). Then, \( \varphi(t) \) is modeled as

\[
\varphi(t) = \begin{cases} 
\varphi_0 & t_0 \leq t \leq t_1 \\
At + B & t_1 \leq t \leq t_2 \\
\varphi_{\text{max}} & t_2 \leq t \leq T 
\end{cases}
\]  
(4.1)

where \( T \) is the total time for which the leg gets the signal from the control chip telling it to move backwards, \( \varphi_0 = \varphi(t_0) \),

\[
A = \frac{\varphi_{\text{max}} - \varphi_0}{t_2 - t_1}
\]

and

\[
B = \varphi_{\text{max}} \frac{1 - \frac{\varphi_0}{\varphi_{\text{max}}}}{1 - t_1/t_2}
\]

Furthermore, we assume that the velocity is zero during \( t_0 \leq t < t_1 \) and \( t_2 < t \leq T \) and jumps to a constant value at \( t_1 \) which is maintained until \( t_2 \), when the velocity jumps back to zero. The values of \( t_1 \) and \( t_2 \) are determined from the data.

**Simplification 3:** The movement of shoulder number \( i \) with respect to foot num-
Figure 4.14: Forward walk, $P \approx 0.89$ s, Duty cycle $\approx 50\%$. Top graph: Movement of the center of gravity in x-direction, $c_x(t)$, as a function of time. Dots with error bars: data; solid line: least squares fit. Middle graph: The dots are the values of the fit subtracted from value of the data as a function of time which lie within the error (lines). Lower graph: Control voltages as a function of time. In the upper row, dots correspond to leg LF, circles to leg RF, and in the lower row, dots correspond to leg RH and circles to leg LH. Displayed symbols indicate that the corresponding leg is active. From the fit displayed in the top graph, $f = 0.7$ is obtained.

The shoulder movement is modeled by $a \sin(\varphi_i(t))$ for $i \in \{1, \ldots, 4\}$ (in x'-direction). In the present implementation, $\varphi_{\text{max}} = 20^\circ$. We can approximate $\sin(\varphi_i(t))$ with $\varphi_i(t)$ without making too large an error. Then, the shoulder movement with respect to the foot is modeled by $a \varphi_i(t)$ for $i \in \{1, \ldots, 4\}$.

### 4.3.1 Forward motion

Assume that active feet (i.e. those which have the electromagnet turned on) stick to the ground without slipping and that feet, when they are not active, slip frictionless over the ground. Then, the velocity of shoulder number $i$ of the active leg number $i$ with respect to the ground is constrained such that the sideways velocity of the shoulder (i.e. the velocity that is at right angles with the x'-axis) must be zero.
The shoulder velocity in the direction of motion of the robot’s body is \( a \dot{\varphi}_i(t) \bar{e}_x \) for \( t_1 \leq t \leq t_2 \) and \( i \in I \), where \( I \) is the set of those subscripts that denote active legs.

If all active legs move with the same velocity, \( \dot{\varphi}_i(t) = \dot{v} \), then the robot moves forward in a straight line and the movement of its center of gravity is given by

\[
c(t) = (avt + c_0)(\cos(\phi_0)\bar{e}_x + \sin(\phi_0)\bar{e}_y)
\]

where \( c_0 \) is the location of the center of gravity and \( \phi_0 \) is the angle between the direction of movement of the robot and the x-axis, at the time when the movement starts.

For walks in which each leg has a duty cycle of 50%, there are always two legs active at the same time, moving with the same velocity and thus the robot can be expected to move according to (4.2), since ideally the total torque is zero. The data shows that the robot’s body indeed moves on a straight line (see Fig. 4.14). But, since there is friction and slip in the real system, the slope, i.e. the velocity, is different from the one predicted in Fig. 4.2. The difference is given by a term \( f = \text{measured velocity}/\text{predicted velocity} \).

### 4.3.2 Walking on a curved trajectory

Driving two legs of the robot with a duty cycle other than 50% causes the robot to turn. Data in Fig. 4.16 shows how the robot turns. The front legs move normally with approximately equally long stance and swing phases and the movement in each phase spans approximately the whole range of the leg, just briefly (if at all) touching the mechanical stops. The hind legs are driven with control signals with a duty cycle not equal to 50%. Their movement is reduced because for one part of the cycle the control signal tells the motor to move in one direction for a time longer than it can, given the mechanical stop. The leg is forced to remain at the mechanical stop until the control signal changes. For the other part of the cycle, the signal tells the motor to move in the other direction for a shorter length of time than it would take the leg to swing over the full possible angle. Thus, the amplitudes of both hind legs’ movements are reduced. During the time that the right front leg is the only active leg, one of the hind legs moves. During part of the time that the left front leg is the only active leg, the movements of both hind legs are obstructed by the mechanical stops. Their motors stall and draw a large current. The four motors are connected in parallel to the same power supply which is set to a current limit of 0.84A. The motor of the left front leg is limited in its power consumption because of the large stall current the two hind motors draw. Thus, the velocities during the active movement phase of the right front leg and the left front leg differ (see Fig. 4.18). The right front leg moves with the higher velocity.

The robot turns left when the right front leg is active. While doing so, the orientation of the robot with respect to the x-axis changes as well as the x- and
Figure 4.15: Sketch of the model of the robot as the body is turning due to shoulders 1 and 2 moving with different velocities. Projection onto the (x,y) plane. World coordinates (x,y), body centered coordinates, (x',y'), with origin at the center of gravity (\(\bar{c}\)) of the robot's body. Front Shoulders (\(\bar{S}_i\), \(i = 1, 2\)). Width (w) and length (l) of the robot's body. The angle \(\phi\) is the angle between x axis and x' axis.
Figure 4.16: Turning. Upper two sub-plots: Voltage of the potentiometers normalized to the interval [0, 1] as a function of time for the four legs: left front (LF; dots), right front (RF; circles), left hind (LH; crosses) and right hind (RH; stars). The x-coordinate (third sub-plot) and the y-coordinate (fourth sub-plot) of the center of gravity are plotted as a function of time. Lower sub-plot: the angle between the longitudinal axis through the body and the x-axis as a function of time, 'angl. displ.' means angular displacement. In sub-plots three to five (counted from the top), stars with error bars are measured data, the line is the fit according to the model described in the text.
Figure 4.17: Strong Turning. Upper two sub-plots: Voltage of the potentiometers normalized to the interval $[0, 1]$ as a function of time for the four legs: left front (LF; dots), right front (RF; circles), left hind (LH; crosses) and right hind (RH; stars). The x-coordinate (third sub-plot) and the y-coordinate (fourth sub-plot) of the center of gravity are plotted as a function of time. Lower sub-plot: the angle between the longitudinal axis through the body and the x-axis as a function of time, ‘angl. displ.’ means angular displacement. In sub-plots three to five (counted from the top), stars with error bars are measured data, the line is the fit according to the model described in the text.
Figure 4.18: During the movement that leads to turning, legs LF and RF move with different velocities since the motor of the LF leg cannot consume as much current as the motor of the RF leg, because the LF leg moves while both hind legs stall and draw large stall currents.

Figure 4.19: Large curvature: Legs LF and RF move with different velocities since the motor of the LF leg cannot consume as much current as the motor of the RF leg, because it moves while both hind legs stall and draw large stall currents. The difference in slopes is 0.7279
y-coordinate of its center of gravity. During the time that the left front leg is active, first both hind legs move as well; thus the left front leg moves with a velocity comparable to the velocity of the active right front leg. During this time the robot turns to the right. As the hind motors stall, the velocity of the active left front leg decreases and so does the speed at which the robot turns right. In sum, after one cycle of the periodic movement there is a net change in the angle \( \phi \), indicating a turn to the left, as well as in the coordinates for the x- and y-coordinate of the robot’s center of gravity, indicating forward motion.

The turning can be made more pronounced when the duty cycle of the control signals for the hind legs deviates more strongly from 50% (see Fig. 4.17). In this case, the hind legs stall during the entire active phase of the left front leg, yielding a smaller velocity of the whole leg movement (see Fig. 4.19). This results in a smaller amount of turning into the antagonistic direction.

To model this behavior, we neglect the effect of the hind legs. For the front legs, we assume that the movement of both legs is constrained such that their sideways velocity must be zero. We assume furthermore that the shoulder of an active leg moves with a velocity that is given by the velocity of the leg in the same way as discussed for forward motion, while the velocity of the shoulder of a non-active leg must be smaller than this because the leg does not move the body forward.

After making these assumptions, we consider what happens when the shoulders of the front girdle move at the same time with different velocities. As mentioned, the sideways velocities (in y’-direction) of the shoulders must be zero. But, since they move with different velocities and they are connected with a rigid piece of metal, they will move approximately on a circle with angular velocity \( \dot{\phi} \). Consider Fig. 4.15. The right front (RF) leg moves faster than the left front (LF) leg, causing the shoulder \( S_1 \) to move with a velocity \( \dot{S}_1 \) that is smaller than the velocity \( \dot{S}_2 \) of shoulder \( S_2 \). An infinitesimal movement of shoulder \( S_2, dS_2 \), greater than an infinitesimal movement of shoulder \( S_1, dS_1 \). This implies that the shoulders must move on a circle, because the body of the robot is rigid. The center of the circle must lie on a line which goes through both shoulders. While the two shoulders travel the distances \( dS_1 \) and \( dS_2 \), respectively, \( \phi \) changes the infinitesimal amount \( d\phi \). Therefore, \( dS_1 = (r - w/2) \sin(d\phi) \). Since \( d\phi \) is very small, this can be approximated by \( dS_1 = (r - w/2)d\phi \). In the same way, \( dS_2 = (r + w/2)d\phi \). Solving for \( r \) and \( d\phi \) yields

\[
d\phi = \frac{1}{w}(dS_2 - dS_1)
\]

\[
r = \frac{w}{2}(dS_1 + dS_2)
\]

\[
dS_2 - dS_1
\]
Eq. (4.3) implies that
\[
\dot{\phi} = \frac{1}{w} (\dot{\hat{s}}_2 - \dot{\hat{s}}_1)
\] (4.5)

This motion causes the robot's center of gravity to move on a circle with radius
\[ r' = \sqrt{r^2 + l^2/4} \]
and angular velocity \( \dot{\phi} \).
The movement sequence is modeled piecewise, applying the results of this section
for those movement phases when either the LF or RF leg is active (see Fig. 4.16
and Fig. 4.17).

4.4 Summary

The WGC chip successfully controls the walking behavior of a four-legged test
robot, enabling the robot to use all of the standard quadrupedal gaits (as shown in
Fig. 1.1).

A simple, kinematic model of the walking machine predicts the movement of the
robot's body given knowledge of the movement of the robot's legs. This model
explains how the robot executes turning maneuvers.
Chapter 5

Learning Walking Gaits

The Walking Gait Control (WGC) chip, described in Chapter 3, provides a four-legged walking robot with the ability to use different walking gaits by controlling the inter-leg coordination of the robot’s legs (see Chapter 4). The output of the chip depends on a small set of input voltages that determine the walking gait, the direction of the robot and the step frequency of the legs. In order to make autonomous use of its motor abilities, the robot has to learn the set of input values that make the WGC chip produce a desired movement pattern.

The architecture of the robotic system is inspired by the architecture of controllers for locomotion found in animals such as the lamprey where input from the brain to the spinal CPG influences the behavior of the spinal CPG. It is believed that this architecture simplifies the control task for the brain, which then only has to decide when to locomote and set the general level of activity [43] while the CPG solves the problem of coordinating the large number of muscles involved in locomotion.

In a similar way the machine locomotion control task is simplified using the architecture shown in Fig. 5.1 which exploits that the WGC chip transforms a set of numbers into rhythmic control voltages. It is easier to learn these numbers than it is to learn the detailed movement sequence, that leads to a desired walking gait and a desired trajectory.

The robotic system (see Fig. 5.2) consists of: (i) the mechanical platform with motors and sensors (as described in Chapter 4), (ii) the WGC chip (as described in Chapter 3) and (iii) a PC on which algorithms are run to (a) acquire data from chip and sensors, (b) set the chip’s input parameters and (c) implement the learning (as described in this chapter).

In principle, it would be possible to build in the knowledge a priori rather than learning it. However, this would be very time consuming, because a look-up table would have to be created (see Sec. 3.4.1) for which the parameter space given by the input voltages would have to be finely scanned; more finely than what is required to
Figure 5.1: Sketch of (A) the (extremely simplified) control architecture for locomotion of vertebrates and (B) the (simplified) control architecture of the robot.

Figure 5.2: Sketch of the control architecture of the robot. Thick arrows indicate the learning loop.
generate the data for the learning, as the particular learning algorithm used here is able to generalize from few examples. Furthermore, the values for the look-up table would have to be obtained with reasonably high precision. Since the data acquisition is noisy (see Sec. 3.2), the output of the chip would have to be measured many times and averaged at each point in parameter space. The learning algorithm can handle noisy data (see Sec. 5.1.2). Another argument in favor of learning is that due to inter-chip variability, the time consuming procedure of creating a look-up table would have to be repeated for every new chip. Finally, once a successful learning procedure is determined, the motor abilities of the system can be exploited by the system itself, independent of the particular physical shape of the walking robot.

5.1 Learning procedure

The problem of learning how to set the input of the WGC chip in order to obtain a desired movement pattern can be reformulated: Given examples, learn the regime(s) in input parameter space which contain parameters that lead to a desired walking gait. This problem can be decomposed into (i) acquisition of the examples and (ii) classification of the examples.

In order to locomote in a specific direction using a specific walking gait, the robotic system must learn to which values to set the $N$ input voltages of its WGC chip ($N$ depends on the configuration the chip is used in). In the present implementation the chip is used in the configuration of a chain of four oscillators as described and discussed in Sec. 3.4, in which $N = 14$. A gait is defined by the phase shifts between the oscillators, $\phi_i$, $(i = 1, \ldots, 3)$ as discussed in Sec. 3.4.1. The direction of the robot is given by the duty cycles, $D_j$, $(j = 1, \ldots, 4)$ of the oscillators (see Chapters 3 and 4). The specifications of the desired movement sequence, $\{\phi_i\}$ and $\{D_j\}$ $(i = 1, \ldots, 3$ and $j = 1, \ldots, 4)$, are given to the system. Eight of the fourteen input voltages are fixed, namely the bias voltages for the transistors $M_1$ on each side of each oscillator (see Figs. 3.4 and 3.6). As a result, both, the frequency of the oscillators, $\nu_j = \tilde{\nu}$ (for $j = 1, \ldots, 4$), and the duty cycle of the first oscillator, $D_1$, are fixed. The step frequency of the robot’s four legs is fixed via $\tilde{\nu}$ to a value that takes the mechanic stops which constrain the amplitude of the leg movement into account. Once we have fixed $\tilde{\nu}$, it is easy to set $D_1$, because it then depends only on one further parameter (see Sec. 5.3). However, $\tilde{\nu}$ and $D_1$ could also be learned if necessary.

If the remaining six input voltages had to be learned together, a region in a six dimensional space would have to be classified. The acquisition of the examples is simplified by learning regions in three two-dimensional spaces instead. This can be done because the phase shift $\phi_i$ is only determined by the input voltages, $V_{b,i,i+1,l}$ and $V_{bi,i+1,r}$.

1The notation is explained in Chapter 3.
and duty cycles can be learned in sequence: first the values for $V_{b,12,l}$ and $V_{b,12,r}$ that result in $\bar{\phi}_1$ and $\bar{D}_2$ are learned. Then the values for $V_{b,23,l}$ and $V_{b,23,r}$ that yield $\bar{\phi}_2$ and $\bar{D}_3$ are learned and finally the values for $V_{b,34,l}$ and $V_{b,34,r}$ which give $\bar{\phi}_3$ and $\bar{D}_4$. Since the learning procedure is the same for each session, in the following the indices $b, i$ and $i + 1$ are dropped from the notation. For each learning session, the two voltages that span the input space are therefore $V_r$ and $V_l$. A point in this space is denoted by

$$V_i = (V_r, V_l), \quad i \in \mathbb{N}$$

### 5.1.1 Creating the training data for the learning algorithm

Let us assume that $\bar{\phi}_1$ and $\bar{D}_2$ are the desired values to be learned. To learn which regime in the two dimensional input voltage space leads to these desired values, the robotic system first tries out different points in the input parameter space. An algorithm sets the input parameters of the WGC chip to some value $V_1$. The data acquisition algorithm records the activity of the WGC chip and computes the output frequencies $\nu_j$ ($j = 1, \ldots, 4$), duty cycles $D_j$ and the phase shifts $\phi_i$ ($i = 1, \ldots, 3$) between the waveforms. The algorithm that generates the training data set compares the measured values of $\phi_1$ and $D_2$ to $\bar{\phi}_1$ and $\bar{D}_2$. If $\phi_1$ and $D_2$ lie within pre-specified tolerances $d\phi_1$ and $dD_2$ around $\bar{\phi}_1$ and $\bar{D}_2$, the point $V_1$ is added to the training data set. This procedure (see Sec. 5.2) is repeated for a set of $m$ points in a subset $X \subset \mathbb{R}^2$ of the input voltage space within reasonable values as known from the analysis of the chip (see Chapter 3). The set $X$ is chosen square such that

$$\max_{i=1,\ldots,m} ((V_i)_i) - \min_{i=1,\ldots,m} ((V_i)_i) = \max_{i=1,\ldots,m} ((V_r)_i) - \min_{i=1,\ldots,m} ((V_r)_i) =: V$$

Using this method, a set of $l$ data points $\{V_i : V_i \in X, i \in \{1, \ldots, l\}\}$ is obtained. This is scaled to the domain $[-1, 1]$ with the transformation $v_i = 2(V_i - V_{min})/V - 1$, where $V_{min} = \min_{i=1,\ldots,m}(V_i)$. Finally the training data set containing $l$ unlabelled examples is given by $\{v_i : v_i \in V, i \in \{1, \ldots, l\}\}$ with $V \subset [-1, 1]^2$.

### 5.1.2 Learning algorithm

A classification algorithm is used to learn those regimes in the WGC chip's input parameter space that yield the desired walking gaits given the training data. The learning algorithm was introduced recently by Schölkopf et al. [103]. It extends ideas developed for binary classification in the context of Support Vector learning

\footnote{For some values of the input voltages period doubling can occur (see Chapter 3). Thus, it has to be checked for every point in input space if the frequencies of all the output waveforms are the same within some tolerance that is given by the noise of the data acquisition. Otherwise the point is disregarded.}
to classification of unlabeled data. To our knowledge, the present implementation is the first application of this algorithm to a robotic system.

The acquisition of the WGC output is affected by noise (see Sec. 3.2) and thus the learning algorithm only has access to noisy data. In a natural environment the robot should be able to learn from few trials. The learning algorithm used (see Sec. 5.1.2) is tailored to exactly this problem: given (potentially few) examples, a subset of the data space can be estimated such that the probability that a data point lies outside of this subset is bounded by some a priori specified value. This way, the algorithm can deal with noisy data.

Learning from unlabeled examples

The learning algorithm [103] computes a binary function, a decision function $f$, which is +1 in the region in input space where (most of) the data lies and -1 outside of this region. In the present application, the training data set lies in a two dimensional space. The boundary given by the decision function is not necessarily a straight line. Thus the classification cannot be done with a linear estimator. A nonlinear learning machine is constructed by first applying a nonlinear mapping to the data, which transforms the data into a space in which it is linearly separable, and then separating the data in this space with a hyperplane. From a representation of this hyperplane the required nonlinear separating curve is computed. The mapping is referred to as a feature map

$$\Phi : \mathcal{V} \mapsto \mathcal{F}$$

where

$$\mathcal{F} = \{ \Phi(\mathbf{v}) : \mathbf{v} \in \mathcal{V} \}$$

is called the feature space [123].

Finding an appropriate mapping $\Phi$ is a nontrivial problem. Fortunately, for the SV learning algorithm it is not necessary to explicitly know $\Phi$ and $\mathcal{F}$ in order to construct the optimal separating hyperplane in $\mathcal{F}$ and determine the decision function. As we will see later, the evaluation of the decision function does not require the evaluation of the mapped input patterns, $\Phi(\mathbf{v})$, in explicit form, but only the evaluation of the dot product $(\Phi(\mathbf{v}) \cdot \Phi(\mathbf{v}))$ where $\mathbf{v}$ and $\tilde{\mathbf{v}}$ are vectors in input space. This is important because the dot product is assumed to be computable by evaluating a kernel $k$

$$k(\mathbf{v}, \tilde{\mathbf{v}}) = (\Phi(\mathbf{v}) \cdot \Phi(\tilde{\mathbf{v}})),$$  \hspace{1cm} (5.1)$$

If a kernel fulfills Mercer's condition [81] (i.e. if $k$ is a continuous symmetric kernel of a positive integral operator), then a mapping $\Phi$ into a dot product space exists
for which (5.1) is true. In the learning algorithm used here, a Gaussian kernel

\[ k(v, \tilde{v}) = e^{-\|v-\tilde{v}\|^2/2\sigma^2} \]  

(5.2)
is used. The kernel implicitly defines the features with respect to which the representation of the data is changed by applying the mapping \( \Phi \). Thus, the choice of the kernel distinguishes a variety of nonlinear estimators. Note that the algorithm can be used with any Mercer kernel, satisfying eq. (5.1). Furthermore, it should be pointed out that with the choice of a Gaussian kernel, there is still a class of learning machines, distinguished by the width \( \sigma \). It is thus important to chose a value of \( \sigma \) that is appropriate for a given learning task. We will return to this problem later in this section.

In feature space, the learning problem is solved with a linear learning algorithm by finding the optimal hyperplane that separates the training data set from the origin. To construct the optimal hyperplane, let \( x_i = \Phi(v_i) \) be the feature space images of the point \( v_i \) in input space, \( i \in \{1, \ldots, l\} \). This data set is separable, if there exists some \( w \in \mathbb{F} \) such that \( (w \cdot x_i) > 0 \) for \( i \in \{1, \ldots, l\} \).

**Proposition [103]:**

If the data is separable, then there exists a unique supporting hyperplane with the properties that (i) it separates all data from the origin and (2) its distance to the origin is maximal among all such hyperplanes. For any \( \rho > 0 \) it is given by

\[ \min_{w \in \mathbb{F}} \frac{1}{2}\|w\|^2 \quad \text{subject to} \quad (w \cdot x_i) \geq \rho, \; i \in \{1, \ldots, l\}. \]

The reason why minimizing \( ||w|| \) maximizes the margin (distance to the origin) of the hyperplane \( \{ z \in \mathbb{F} : (w \cdot z) = \rho \} \) is that the margin is given by \( \rho/||w|| \). The binary decision function is given by

\[ f(v) = \text{sgn}((w \cdot \Phi(v)) - \rho) \]  

(5.3)

So far, an implicit assumption was made that it is desirable to find a boundary such that *all* the training points lie within the region on which \( f \) is positive. Since the training data is noisy this might lead to misclassifications on test data. To account for the noise in the data, the classification is “softened”. The algorithm is reconstructed such that it tolerates some fraction of outliers. For this purpose, slack variables \( \xi_i \geq 0 \) are included which are penalized in the objective function if they are non-zero. The constraints are relaxed to \( (w \cdot \Phi(v)) \geq \rho - \xi_i, \; i \in \{1, \ldots, l\} \).
The constraint optimization problem becomes

\[
\min_{w \in \mathcal{F}, \xi \in \mathbb{R}, \rho \in \mathbb{R}} \frac{1}{2} ||w||^2 + \frac{1}{\nu} \sum_{i=1}^{l} \xi_i - \rho \\
\text{subject to } (w \cdot \Phi(v)) \geq \rho - \xi_i, \xi_i \geq 0
\]

(5.4) (5.5)

The value of \( \nu \) controls the trade-off between the two goals that (i) the decision function (5.3) is positive for as much of the training data as possible while (ii) the term \( ||w|| \) is kept small.

The constraint optimization problem of eqs. (5.4) and (5.5) is a quadratic programme meaning that the objective function is quadratic and all the constraints are linear. Furthermore, the objective function is convex. Convex quadratic programmes can be solved using the Lagrange multiplier technique which was initially developed in 1797 by Lagrange for mechanical problems that had only equality constraints. The method was extended to inequality constraints by Karush [63] and by Kuhn and Tucker in 1951 [70].

The Lagrangian for eqs. (5.4) and (5.5)) is

\[
L(w, \xi, \rho, \alpha, \beta) = \frac{1}{2} ||w||^2 + \frac{1}{\nu} \sum_{i=1}^{l} \xi_i - \rho \\
- \sum_{i=1}^{l} \alpha_i ((w \cdot \Phi(v_i)) - \rho + \xi_i) - \sum_{i=1}^{l} \beta_i \xi_i
\]

(5.6)

where \( \alpha_i \) and \( \beta_i \) are the Lagrange multipliers. The saddle point conditions are

\[
\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{l} \alpha_i \Phi(v_i)
\]

(5.7)

\[
\frac{\partial L}{\partial \xi} = 0 \rightarrow \alpha_i = \frac{1}{\nu} - \beta_i \leq \frac{1}{\nu}
\]

(5.8)

and

\[
\frac{\partial L}{\partial \rho} = 0 \rightarrow \sum_{i=1}^{l} \alpha_i = 1
\]

(5.9)

The points \( \{v_i : i \in \{1, \ldots, l\}, \alpha_i > 0\} \) are called Support Vectors. They are the only points occurring in the expansion of the hyperplane (5.7). Points with \( \xi_i > 0 \) are called outliers, since they lie outside of the estimated region.

It can be shown ([103], Proposition 4) that \( \nu \) is an upper bound on the fraction of outliers and a lower bound on the fraction of Support Vectors. As the choice of \( \nu \)
determines the behavior of the classifier, it has to be chosen carefully. This will be discussed later together with criteria for choosing $\sigma$.

Substitution of (5.7) into (5.3), using (5.1), yields a kernel expansion of the decision function

$$f(v) = \text{sgn} \left( \sum_{i=1}^{t} \alpha_i k(v_i, v) - \rho \right)$$  \hfill (5.10)

The Lagrange method leads to an alternative dual problem that results from the introduction of the Lagrange multipliers (or dual variables). It is often easier to solve than the original problem [25]. The idea is that the dual variables are the fundamental unknowns of the problem [123]. In the present implementation, the quadratic programme is solved with a method that makes use of the dual description of the problem. The dual problem is obtained by substitution of eqs. (5.7) to (5.9) into (5.6), using (5.1):

$$\min_{\alpha} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j k(v_i, v_j)$$

subject to $0 \leq \alpha_i \leq \frac{1}{\nu_i}$, $\sum_{i=1}^{t} \alpha_i = 1$  \hfill (5.11)

It can be shown [103] that if $\alpha_i$ and $\beta_i$ are nonzero, the inequality constraints (5.5) become equalities at the optimum, which allows to recover $\rho$, as for any $0 \leq \alpha_i \leq \frac{1}{\nu_i}$, the corresponding $v_i$ satisfies (5.7):

$$\rho = (w \cdot \Phi(v_i)) = \sum_{j} \alpha_j k(v_j, v_i)$$

An inequality constraint (as in eq. (5.5)) is said to be active if for the solution of the optimization problem, $w^*$, equality is satisfied. According to the Karush-Kuhn-Tucker complementary conditions [70], $\alpha_i > 0$ is only possible for active constraints while $\alpha_i = 0$ is necessarily the case for inactive constraints. Thus the Support Vectors are the only points with active constraints. If $v^*$ is a Support Vector, then $(w^* \cdot \Phi(v^*)) = \rho$, which means that the Support Vectors lie on the decision boundary. To be able to classify them as belonging to the estimated region, the obtained value of $\rho$ is slightly offset in the implementation below by multiplying it with a constant $\eta$ which is smaller than (but close to) 1.

It is now clear that $f$ can be computed without explicit knowledge of the feature map $\Phi$ (compare eq. (5.10)).

The choice of the parameters $\sigma$ and $\nu$ is nontrivial and is so far been solved heuristically [103]. We argue that the variance of the Gaussian kernel should scale with the distance of the nearest neighbors in the training data set. We chose $\sigma$ to be close to the minimum of the distances between nearest neighbors in the training data.
The fine-tuning of $\sigma$ is done heuristically. The parameter $\nu$ controls the fraction of outliers, which is related to the noise that the training data is subject to. $\nu$ is adjusted to a value which makes the algorithm disregard approximately as many points as can be expected to be falsely included in the training data given the noise of the data acquisition.

After learning, the robotic system has an estimate of the region in which input voltages lie that cause the WGC chip to produce the desired walking gait. The robotic system thus learns a representation of desired movement sequences. It can now set the input to its WGC chip accordingly whenever it needs to use this gait.

5.2 Implementation

The algorithmic implementation of the learning procedure is mostly written in C (steps 1 to 7, 13 and 14), interfaced to the learning algorithm’s numerical implementation. The latter is a Matlab program originating from code produced by Schölkopf which was modified for the present application. To determine the phase shifts $\phi_i$ and the duty cycles $D_{i+1}$, the correct values for $V_{b,i+i,j}$ and $V_{b,i+i+1,r}$ are learned (here and in the rest of this section $i = 1, \ldots, 3$). The user chooses values $\{\phi_i\}$ and $\{D_j\}$ ($i = 1, \ldots, 3$ and $j = 1, \ldots, 4$) which specify the desired gait, tolerances $\{d\phi_i\}$ and $\{dD_j\}$, the input voltage space that is going to be scanned, $X$, the scanning step size and the number of times, $n$, over which the measurement at each point in input voltage space is averaged. The implemented procedure is:

1. $q \leftarrow 1$. While $q \leq 3$ do the following:

2. Set the input voltages to the WGC chip.

3. Acquire the output waveforms of the WGC chip.

4. Compute the frequencies, $\nu_j$, and duty cycles, $D_j$ of each output and the phase lags between outputs, $\phi_i$ ($j = 1, \ldots, 4$ and $i = 1, \ldots, 3$). Every value is obtained by averaging $n$ measurements taken at cycles adjacent in time.

5. IF $\bar{D}_j - d\bar{D}_j \leq D_j < \bar{D}_j + d\bar{D}_j$ AND $\bar{\phi}_i - d\bar{\phi}_i \leq \phi_i \leq \bar{\phi}_i + d\bar{\phi}_i$ AND $\nu_j = \bar{\nu}$ ($\forall i \in \{1, \ldots, 3\}$ and $j \in \{1, \ldots, 4\}$) THEN add the vector consisting of the current values of $V_{b,i+i,j}$ and $V_{b,i+i+1,r}$ to the training data set.

6. $V_{b,i+i+1,t}$ and $V_{b,i+i+1,r}$, are changed according to the specifications given by the user and set.

7. Repeat (starting at 2) until all of $X$ is scanned.

8. Scale the training data set to $[-1, 1]^2$. 
9. Determine \( \alpha \) by solving the quadratic program according to eqs. (5.4), (5.5), (5.11) and (5.12).

10. Evaluate \( \rho \) and the decision function \( f \) (compare eqs. (5.13) and (5.10)) (Note that \( \rho \) is slightly offset).

11. Compute the training error.

12. Evaluate the argument of \( f \) for points which lie on a (fine) grid that discretizes the input space. Compute \( f \) on this grid and display where the argument of \( f \) is zero.

13. Choose a point close to the geometrical center of the estimated region. The value of the voltages \( V_{b,1,i} \) and \( V_{b,1,i+1} \) at this point are used from now on.

14. \( q \leftarrow q + 1 \). Repeat starting at 2.

The software package LOQO, code for quadratic programming, is used in step 9 to solve the optimization problem implementing a primal-dual interior-point method (documented in detail in [122] and [121]).

5.3 Learning different movement sequences

We first focus on letting the machine learn to walk forward using different walking gaits. For the robot to move roughly on a straight line, all oscillators need to have a duty cycle of roughly 50% (see Chapter 3). The duty cycle of the first oscillator
can theoretically be set to 50% by setting $V_{b,1,l} = V_{b,1,r}$. In practice, the two values will be slightly different, because of mismatches between the two transistors, but the value of $V_{b,1,l}/V_{b,1,r}$ that yields $D_1 = 0.5$ can easily be measured (see Fig. 5.3). In the following, $V_{b,i,l}$ and $V_{b,i,r}$ are preset to $V_{b,i,l} = V_{b,1,l}$ and $V_{b,i,r} = V_{b,1,r}$ for $i = 2, 3, 4$, if not stated otherwise.

This leaves us with six remaining unknown parameters (see Sec. 5.1):

$V_{b,1,l}$ and $V_{b,1,r}$ set $\phi_1$ and $D_1, D_2, D_3$

$V_{b,23,l}$ and $V_{b,23,r}$ set $\phi_2$ and $D_2, D_3$

$V_{b,34,l}$ and $V_{b,34,r}$ set $\phi_3$ and $D_3$

In many gaits one or more of the phase shifts are 0.5. The parameters that generate a phase shift of 0.5 need not be learned, because this can be achieved trivially, with the particular implementation used here, by setting the gate voltage of the transmission gates which couple two oscillators (see Sec. 3.4.1) to a voltage such that both transistors are turned on and current flows through the transmission gate.

Since phase shifts of 0.5 can be produced trivially, the bound gait, which is defined by $\phi_1 = \phi_2 = \phi_3 = 0.5$ (because the chip is configured as in Fig. 3.13) is a gait which need not be learned. There are three gaits which have only phase shifts of either 0.5 or 0, namely the trot, the pace and the pronk. The simplest way to produce a phase shift of $\phi = 0$ is to use a relay to switch the polarity of the motor. The signal which determines the state of the relays is an additional control signal that is set by the computer.

After these simplifications, the gaits that need to be learned are the walk gait, the canter gait, the transverse gallop and the rotary gallop (see Fig. 3.4.1). We show results for two examples, the walk gait and the gallop. Note that these gaits are only examples for movement sequences which can be controlled by the WGC chip and that it is possible to learn any leg sequence with the procedure described in this chapter. For the walk gait we need to learn $\phi_1 = \phi_2 = \phi_3 = 0.75$ and for the transverse gallop, $\phi_1 = \phi_3 = 0.5$ can be set while $\phi_2 = 0.6$ has to be learned.

### 5.3.1 The Walk Gait

The input parameters are learned for a forward walk, requiring phase shifts of $\phi_1 = \phi_2 = \phi_3 = 0.75$ and duty cycles of $D_1 = D_2 = D_3 = D_4 = 0.5$. The oscillation period $P = 0.89s$ and the duty cycle $D_1 = 0.5$ are fixed. The scanning step size is 2mV and the tolerances are chosen to be ±0.015 for the phase lags and ±0.05 for the duty cycles. The parameters $V_{b,j,j+1,l}$ and $V_{b,j,j+1,r}$ are learned in sequence, first for $j = 1$ (see Fig. 5.4). The result is applied to the WGC. Then $V_{b,23,l}$ and $V_{b,23,r}$

The notation is explained in Chapter 3.
Figure 5.4: Result of learning values of the bias voltages $V_{b,12,l}$ and $V_{b,12,r}$ which lead to $\phi_1 = 0.75$ and $D_2 = 0.5$. Correctly classified (stars) and misclassified (crosses) training data (a) and test data (b). Outlined regions: learned by algorithm from training data. The training data is obtained from one scan (as explained in the text). The test data is a set of points obtained from three scans. $\nu = 0.2, \sigma = 0.23$.
Figure 5.5: Result of learning values of the bias voltages $V_{b,23,l}$ and $V_{b,23,r}$ which lead to $\phi_2 = 0.75$ and $D_3 = 0.5$, while $\phi_1 = 0.75$ and $D_2 = 0.5$ are fixed using the results from Fig. 5.4. Correctly classified (stars) and misclassified (crosses) training data (a) and test data (b). Outlined regions: learned by algorithm from training data. The training data is obtained from one scan (as explained in the text) as well as the test data. $\nu = 0.2$, $\sigma = 0.23$. 
Figure 5.6: Result of learning values of the bias voltages \( V_{b,34,l} \) and \( V_{b,34,r} \) which lead to \( \phi_3 = 0.75 \) and \( D_4 = 0.5 \) while \( \phi_1 = \phi_2 = 0.75 \) and \( D_2 = D_3 = 0.5 \) are fixed using the results from Figs. 5.4 and 5.5. Correctly classified (stars) and misclassified (crosses) training data (a) and test data (b). Outlined regions: learned by algorithm from training data. The training data is obtained from one scan (as explained in the text) as well as the test data. \( \nu = 0.14, \sigma = 0.23 \)
Figure 5.7: Result of learning values of the bias voltages $V_{b,23,l}$ and $V_{b,23,r}$ which lead to $\phi_2 = 0.6$ and $D_3 = 0.5$. Stars: training data (A) and test data (B). Outlined regions: learned by algorithm from training data. $\nu = 0.4$, $\sigma = 0.28$

are learned and the result is also applied to the WGC. Finally, $V_{b,34,l}$ and $V_{b,34,r}$ are learned. The robotic system applies the result of the learning to its WGC and uses a walk gait (the data is shown in Fig. 4.3) to move forward (see Fig. 4.14).

5.3.2 The Transverse Gallop

In the same way as described above, the phase shift $\phi_2 = 0.6$ for the transverse gallop (see Fig. 4.6) can be learned (see Fig. 5.7).

5.3.3 Walking on a curved trajectory

According to Chapter 4, the robot turns when the duty cycles of the hind legs are 30%. The parameters which generate the data for the turning maneuver of Fig. 4.16 are learned with the strategy described above. The result of the learning is shown in Fig. 5.8 and Fig. 5.9.

5.4 Summary

The learning procedure we have described in this chapter uses an unsupervised Support Vector learning algorithm to classify those regions in the input parameter space of the WGC chip which lead to a desired gait. With the procedure described
Figure 5.8: Result of learning values of the bias voltages $V_{b,23,d}$ and $V_{b,23,r}$ which lead to $\phi_2 = 0.75$ and $D_3 = 0.3$, while $\phi_1 = 0.75$ and $D_2 = 0.5$ are fixed using the results from Fig. 5.4. Correctly classified (stars) and misclassified (crosses) training data (a) and test data (b). Outlined regions: learned by algorithm from training data. The training data is obtained from one scan (as explained in the text) as well as the test data. $\nu = 0.18$, $\sigma = 0.23$. 

\begin{center}
(a) 
\end{center} 

\begin{center}
(b) 
\end{center}
Figure 5.9: Result of learning values of the bias voltages $V_{b,34,l}$ and $V_{b,34,r}$ which lead to $\phi_3 = 0.75$ and $D_4 = 0.3$ while $\phi_1 = \phi_2 = 0.75$, $D_2 = 0.5$ and $D_3 = 0.3$ are fixed using the results from Figs. 5.4 and 5.8. Correctly classified (stars) and misclassified (crosses) training data (a) and test data (b). Outlined regions: learned by algorithm from training data. The training data is obtained from one scan (as explained in the text) as well as the test data.
here, we have found a method to automatically fine-tune the parameters of an aVLSI chip. In principal, this method could be useful for other neuromorphic chips as well. It would be ideal, to determine the parameters of the Gaussian Kernel function in an automatic way. This is the subject of current research [53].
Chapter 6

Conclusion and Outlook

6.1 Achievements

An existing framework of building simple circuits for controlling walking machines has been analyzed and extended. These circuits are networks of threshold elements that can form networks of coupled oscillators, where each oscillator can be understood as a 'ring' of two such threshold elements. In particular, the pattern generating capabilities of chains and rings of coupled oscillators, and of larger rings of coupled threshold elements, have been analyzed. I have found that chains of coupled oscillators are best suited to control the behavior of walking machines when these particular oscillators are used.

Based on the proceeding analysis, a novel neuromorphic chip to control inter-leg coordination of four-legged walking machines was introduced. The chip contains five oscillators and four coupling circuits and can be used in different configurations. The pattern generating abilities of the chip were analyzed and tested for two biologically interesting architectures of a network of four coupled oscillators. The result was that a chain of four coupled oscillators is the best minimal circuit for inter-leg coordination. The chip was tested in this configuration on a real walking robot which was created for this purpose. Experimental results confirmed that the chip is able to control the robot's walking gaits, as was anticipated from the circuit analysis.

The chip is a compact, low power solution for robotic gait control. In principal, it can act on a variety of walking machines (see Sec. 6.3). The chip is flexible in two ways. First, it can be used in different configurations each of which implements a different network architecture. Several chips can also be interfaced with direct connections to yield larger networks (see Sec. 6.3). Secondly, the chip can receive input from many different types of devices. An example was provided showing that sensory information from simple sensors can yield adaptation of the chip's output via a direct feedback loop from the sensor to the chip.
Furthermore, the chip can receive input from a higher order control system, similarly to the control structures found in biology. The higher order control system was implemented on a PC. The chip coordinated the complex pattern of motor movements necessary to generate locomotion in the robot while algorithms run on the PC set the input parameters of the WGC chip. In this framework, a learning task was implemented in a feedback loop involving the chip and the PC. The problem of fine tuning the input bias voltages of the chip such that a desired output (i.e. walking gait) would be achieved was reduced to a classification problem of unlabeled, noisy data. This problem was solved using an unsupervised Support Vector algorithm running on the PC. This implementation is an example of the utility of interfacing the WGC chip to a PC.

The problem of having to fine tune a set of bias voltages occurs frequently with neuromorphic chips [97]. If such neuromorphic systems are to be truly autonomous machines, i.e. able to function independently of the intervention of the experimenter, it is necessary to automate the parameter tuning. The strategy that was proposed and successfully used here is in general applicable to other chips. If it is possible to characterize and measure the output of the chip, then the method proposed here can be used to determine which input parameter regime leads to a desired output. It is not necessary to scan the input parameter space as was done in the present implementation. Another possibility which might save time is to search the input parameter space randomly. The advantage of the proposed learning strategy over simply creating a look up table is the speed. This should become especially pronounced for high dimensional input parameter spaces. The advantage over simply taking any of the (randomly or systematically) found examples is that the present strategy delivers knowledge of one (or several) connected region(s) in input parameter space that contain parameters which lead to the desired output. By choosing a point close to the center of such a region, the parameter is more likely to remain within the desired region if the voltage drifts slightly (that means a slight shift to a nearby point in input parameter space), and thus the chip produces the desired output independent of such small disturbances.

The hybrid system created by interfacing the WGC chip to a PC is useful both for the development of a second generation of neuromorphic motor control chips able to solve more complex tasks as well as for other applications (see Sec. 6.3). In the specific application presented in this thesis, the learning strategy enabled the robot to learn how to produce a desired gait. Thus, the robotic system was able to exploit the motor abilities of the WGC chip contained in it, without need for human intervention.
6.2 Problems and possible solutions

- The walking robot has the minimal amount of degrees of freedom (DOF) necessary to test how the chip controls the phase relationships between the movements of the legs, namely one DOF per leg. The movement of the legs is thus restricted to a one dimensional movement. The robot can turn despite of this restriction, but the robot’s navigation capabilities would be increased if there were additional DOF. If the robot was to be used on a non-magnetic surface, it would have to move forwards without the electromagnets. The most elegant mechanical solution would be to have three DOF per leg, enabling each leg to lift off the ground, move forwards and backwards as well as rotate. Many walking animals and most elaborate walking robots have three DOF per leg. It is conceivable that the WGC chip could be used on an existing robot with three DOF per leg rather than constructing a new machine of this kind. For this, an extended network of oscillators would have to be used (see Sec. 6.3).

- The only part of the learning procedure that is not automatic is the choice of the parameters of the Gaussian Kernel function which determine the behavior of the Support Vector learning algorithm. This is a general problem of Support Vector machines and is subject to current research [53].

- A feature of the WGC chip is that the coupling is unidirectional and thus two parameters set one phase lag. This can be advantageous also for multi-joint control, especially if one wants to implement goal-directed behavior because it offers the possibility of individual control of the phase relationships between all the joints. Note that for goal-directed behavior these can differ from case to case. However, to implement symmetrical networks of coupled nonlinear oscillators, as suggested in [36], a different kind of oscillator and a different kind of coupling circuitry would have to be used.

6.3 Possible extensions

As both the theoretical framework developed and the chip are very flexible, the work described in this thesis can be expanded in many ways.

6.3.1 Extending the WGC chip’s control capabilities

Multiple WGC chips could be used to control the rhythmic movements of robots with more than four legs and of robots with multiple jointed legs.

First, the control network could be expanded to control a $2n$ legged robot, ($n \in \mathbb{N}$). Instead of a chain of four coupled oscillators a chain of $2n$ coupled oscillators would be used. Dividing the chain into segments of four oscillators which fit on one chip,
Figure 6.1: Sketch of control architecture for a four-legged robot with three DOF per leg (hip, knee, ankle) using three WGC chips. Round symbols represent oscillators, thick arrows coupling circuits and thin arrows output signals.

[n/2] chips of the same kind could be used. The output of the last oscillator on chip \(i\) would be directly connected to the input of the first oscillator on the chip \(i + 1\), where \(i \in \{1, \ldots, \lfloor n/2 \rfloor\}\).

Secondly, the control network could be expanded to control a robot with \(m\)-jointed limbs, \((m \in \mathbb{N})\). The output of one of the \(2n\) oscillators in the chain would drive one joint of one leg and at the same time would be directly connected to the input of a chain of \(m - 1\) oscillators, making the first oscillator the master of this chain. The resulting network would consist of one chain, containing \(2n\) oscillators, for the inter-leg coordination and \(2n\) chains, containing \(m\) oscillators, for the inter-joint coordination within each limb. An example control architecture for a four-legged robot with three DOF per leg is shown in Fig. 6.1. Note that this kind of architecture is possible because the coupling is unidirectional, affecting only the oscillator which is lower in the hierarchy of the chain.

6.3.2 Extending the system’s learning capabilities

In the learning procedure described in this thesis the output of the chip was measured to determine the robots walking gait. An obvious extension of this work is to use the same procedure but measure the output of the potentiometers in order to determine the behavior of the robot. Under the conditions the learning exper-
iments are subject to, there would be no interesting difference between these two cases. However, under different conditions it could become necessary to consider the sensory information in the learning loop.

The learning algorithm we used could also be employed to make a robot learn about other sensory input. The following example should illustrate this: A neuromorphic tracker chip [59] finds the most salient (highest contrast) region in the image it sees and provides three analog voltage outputs, one of which encodes the target position, the second voltage encodes the target's horizontal speed to the left and the third voltage encodes the horizontal speed to the right. Assume that the machine should execute a tracking task and track the highest contrast object in its field of view. It could do this by learning with which behavior to react to sensory input in order to optimize the tracking task. A possible implementation would be the following procedure. Give the robot a set of stereotyped behaviors with which it can react, such as: fast forward locomotion using a trot gait, turning left with two different curvatures and turning right with two different curvatures. For each behavior, create examples by measuring the output of the tracker chip at time $t$, applying the behavior and measuring the output of the tracker chip again at time $t+1$. If at time $t+1$, the target is closer to the center than at time $t$, then the point in the tracker-output space that was recorded at time $t$ is added to the training data set which is used to train the SV learning algorithm.

We have used prior knowledge obtained from the circuit analysis to speed up the procedure. However, in principal this is not necessary.

6.3.3 Extending the systems sensory input

The WGC chip could be interfaced to many other neuromorphic chips, yielding complex robotic systems. The following two examples could be realized in a straightforward way using the existing components. First, Lewis et al. [72] recently presented a neuromorphic chip that controls the velocity of the leg movement. They use an adaptive motor neuron output stage which controls the leg's motor and which receives input from an oscillator with the same output characteristics as the oscillators on the WGC chip, i.e. square wave voltage. Thus, their adaptive motor neuron circuit could be directly interfaced to the WGC chip. The resulting system would combine gait control with adaptive velocity control. Secondly the WGC chip could be interfaced with neuromorphic sensors, which have been around for many years [80], extracting cues from a visual scene. Consider the simplest case of two behaviors, a fast trot gait and a slow walking gait. Thresholding the average output of a 2 dimensional retina chip would provide information about the density of edges in a visual scene and be sufficient to chose between these two behaviors. This would be a first step towards making the robot select an appropriate walking gait depending on the terrain. Terrains with many objects can be expected to have a high edge density. In these terrains a slow, flexible gait is most adequate. Terrains
with few objects can be expected to have a low edge density, enabling the robot to use a faster gait.

Finally, if one aims at building a walking machine that is easy to control and that is to be used for some specific application, then the chip introduced here is useful because the distinct sets of control parameters it needs as input to generate a certain rhythmic movement sequence could be delivered by a simple converter that takes, for example, input from switches which indicates the gait chosen by the user of the machine and input from a steering device operated by the user.

If, on the other hand, the goal is to build a fully autonomous walking machine that can handle rough terrain and is controlled by neuromorphic hardware, it is certainly necessary to include more sensory processing. The system described in this thesis can be used as a tool for the development of new chips that address this issue. The system provides an interface between PC and walking robot. Because of the chip fabrication time and costs, it would be reasonable to simulate new implementations using the robotic system introduced in this thesis, before building a new chip.
Bibliography


This thesis was supervised by Prof. Rodney Douglas whom I thank for helpful conversations, for valuable comments on the final draft of the thesis and for generous provision of equipment. Prof. Klaus Hepp co-supervised my thesis and helped with many insightful discussions, particularly with advice concerning Chapters 4 and 5. I am very grateful for his encouragement and support.

Dr. Misha Mahowald introduced me to neuromorphic engineering during the first year of my thesis research. She was an excellent teacher and advisor. It is impossible for me to find appropriate words that express how much she contributed to my life with her interesting thoughts and her good, warm-hearted personality. Her tragic, sudden death remains a deep shock.

For financial support I thank the CSEM, Neuchâtel, the Physics Department of the ETH Zürich and the SPP of the Swiss National Science Foundation.

Special thanks go to Adrian Whatley, Dr. Giacomo Indiveri and Dr. Shih-Chii Liu. They helped in so many ways and always had time for me. Adrian Whatley proof read the thesis, wrote the C-code for the tracking algorithm used in Chapter 4 and has helped me to improve both my programming style and my English conversation skills. It is hard to imagine not having him sit in the same office anymore. I would like to thank him for his friendship and his really cool humor! Dr. Indiveri and Dr. Liu read large parts of the thesis and particularly helped with discussions about Chapter 3. I feel much obliged to them for valuable advice on scientific and other matters.

The circuits analyzed in Chapter 2 originate from Mark Tilden. I thank him for important conversations about his work and for his hospitality while I visited Los Alamos National Labs. I am grateful to Prof. Gert Cauwenbergs for helpful discussions concerning the design of the chip presented in Chapter 3. Armin Kühne is acknowledged for his friendly assistance in the machine shop. He fabricated most of the mechanical parts of the robot described in Chapter 4.

I would like to thank all of the members of the Institute of Neuroinformatics (INI) for contributing to a good working atmosphere. Dr. Jörg Kramer and Dr. Tobi Delbrück helped me many times and answered numerous questions for which I am very grateful. Dr. Virginia Meskenaite, Dr. Bashir Ahmed, Dr. John Allison and John Anderson gave me insights into neuroscience by introducing me to their work when I first joined the INI. I thank Dr. Ben Arthur for interesting, scientific discussions and also for his great humor. Dr. Astrid von Stein, Dr. Johannes Sarnthein and their two wonderful sons (especially my great little love Felix) reminded me that there is more to life than just science. I'm very grateful to Prof. Marie-Claude
Hepp-Reymond for her friendship and moral support. Thanks go to Wouter Brok for reading Chapter 2 and for spreading a good-humored attitude. Felix Hürlimann and Dr. Tobe Freemann were great swimming buddies and their refreshing craziness added much fun to my life. Special thanks go to Dr. Philipp Häfliger and Daniel Blank for being good office mates and to Dr. Tom Binzegger.

I was fortunate to meet many interesting people at conferences and workshops. Most of all, I thank Dr. Bernhard Schölkopf for his deep understanding, his friendship and for many inspiring conversations. The original matlab code for the Support Vector algorithm, which was modified for the present application, was provided by him. Additionally, I thank him for useful comments on Chapter 5. I am extremely grateful to Prof. William Bialek for very inspiring discussions and for making me see a light at the end of the PhD-tunnel. I thank Dr. Gwendal LeMasson and Dr. Catherine Breslin for fruitful collaborations. Furthermore, for a variety of reasons, including inspiring scientific conversations, warm thanks go to Dr. Udo Ernst, Dr. Kai Bongs, Olaf Reinecke, Dr. Rogene Eichler-West, Prof. Jim Bower and his wife Carolina, Dr. Ille Gebeshuber, Dr. Karsten Bromann, Prof. Andreas Andreou, Prof. Tim Horiuchi, Dr. Girish Patel, Dr. Anthony Lewis, Matt Marjanovic, Gayle Wittenberg, Silvio Bohrer, Bri Römmer, Dr. Jörg Wehmeier and Dr. Erich Zulauf.

Thanks to all the participants of the 1996 NSF Telluride Workshop in Neuromorphic engineering and all of the people at the 1999 Woods Hole Workshop ‘Methods in Computational Neuroscience’ for making those workshops such intense, inspiring and good experiences.

Last, but not least, I thank my wonderful parents Marlies and Michael Still. Words cannot describe how grateful I am for their abundant love and support. I feel much obliged to Christian Rust for being there whenever I need a friend. I thank Isabel Hepp and Laetitia Gabernet for being great friends, Maria Liebertz, Juppi and Ingrid Kusenbach for their marvelous spirit, and my dear Anna Kovasna, dive and party buddy.
Curriculum Vitae

Personal data
Name
Date of Birth
Place of Birth

Susanne Still
26th July 1970
Langenhagen
(Kreis Hannover), Germany

Education
Doctoral Student at the Institute of Neuroinformatics, Physics Department, ETH Zürich
MBL Course: Methods in Computational Neuroscience at Woods Hole, MA, USA
Crete course in Computational Neuroscience at Heraklion, Crete
Telluride NSF Workshop on Neuromorphic Engineering at Telluride, CO, USA
Diplom in Physics from the University of Hannover, Germany
Student of Physics at the ETH Zürich (EMSφS scholarship)
Student of Physics at the University of Hannover, Germany
Student of Sinology at the University of Berlin (FU)
School Education in Langenhagen, Germany

Nov. 1995 - June 2000
1995
1996 and 1997
1993 - 1994
1990 - 1995
1989-1990
1976-1989