A Control Systems Study of Quadruped Locomotion

RICHARD AF KLINT

Master’s Degree Project
Stockholm, Sweden 2005

TRITA-NA-E05031
A Control Systems Study of Quadruped Locomotion

RICHARD AF KLINT

TRITA-NA-E05031

Master’s Thesis in Computer Science (20 credits) at the School of Electrical Engineering, Royal Institute of Technology year 2005
Supervisor at Nada was Örjan Ekeberg
Examiner was Anders Lansner
A control systems study of quadruped locomotion

Master’s Thesis

Abstract
In this Masters project, I studied the effects of two modifications to a model of quadruped locomotion. This model (the “Robocat”) was developed by Dr. Sergiy Yakovenko at Dr. Arthur Prochazka’s laboratory at the University of Alberta, Canada for the simulation of the neural control of the hindlimbs in a walking cat. I tested the effects of introducing a Fuzzy Logic controller, FLC, to control the transition in CPG-activation between the two phases of locomotion: Stance and Swing. Furthermore, I proposed three ‘global’ reflexes and tested their stability compared to a model in which individual muscle actuators were controlled by stretch reflexes corresponding to length, velocity and force feedback.

The stability of the model setups was assessed by parametric variations of the maximum isometric force of the actuators exerted by central activation. A stability measurement based on a K-nearest-neighbour classification of the parametric variations was then used to quantify the stability. I found that a FLC, with only nine fuzzy rules based on global parameters of the legs, could greatly improve stability and the spread of stable velocities in the Robocat model. In addition, I found no significant change in stability using the ‘global’ reflexes, as compared to the original stretch, which supports the idea that these feedback systems may be interchangeable.

En studie av kontrollsystemen för fyrbent gång

Examensarbete

Sammanfattning

Preface and Acknowledgement

This thesis is the result of half a year research and development within the frame of a Master’s Thesis project of the Royal Institute of Technology (KTH) in Stockholm, Sweden. Through the kind assistance from my supervisor Dr. Örjan Ekeberg, at Studies of Artificial Neural Systems group (SANS), KTH, I was introduced to Dr. Arthur Prochazka and his and Dr. Sergiy Yakovenko simulations on the hindlimbs of walking cats. Dr. Arthur Prochazka invited me over to Edmonton, Canada, where all the simulation work and most of the literature study was done during the fall of 2004. The Master’s Thesis was then finalized in Stockholm, Sweden, during 2005.

I would like to thank all parties involved that have made this Masters Thesis project come true. Especially, I would like to thank Dr. Örjan Ekeberg for introducing me to the fascinating field of locomotor simulations. I appreciate his assistance and interest in this Project. I truly appreciate Dr. Sergiy Yakovenko for letting me use his model and for answering all my trivial questions. Furthermore, I am deeply grateful to Dr. Arthur Prochazka for inviting me over to Edmonton and for being such a great supervisor, always willing to stretch out a helping hand even in the proof reading of this Thesis, and for being such a good friend.
1 Introduction

Throughout evolution, locomotion has been essential for survival. The system controlling this behaviour can therefore be assumed to have a great deal of redundancy. In the last ten years it has become apparent that the reflexes controlling locomotion are task dependent (Prochazka, 1996a). The reflexes previously thought of as hardwired have been shown to vary during the phases in locomotion (Kandel et al., 2000). In insects, as well as in cats, some reflexes have even been shown to change from inhibitory during static posture to excitatory in the stance phase of locomotion (Conway et al., 1987; Orlovsky et al., 1999).

In order to cope with the multivariate nature of sensory feedback researchers have started using conditional logic to describe locomotor control (Prochazka, 1996a). This is expressed in the form of “IF-THEN” rules: IF this sensory state AND that sensory state THEN perform some motor action. Several rules for controlling locomotion have been identified from studies of locomotion in species as diverse as humans, cats, lobsters and locusts (Prochazka, 1996a).

In this Master’s Thesis, I will focus on two of the more basic control mechanisms involved in walking. Firstly, I will consider the effects on postural stability of introducing a Fuzzy Logic controller (FLC) instead of a fixed-rule conditional logic handling the transition of walk phases. Secondly, recent studies, (Bosco et al., 1996; Poppele et al., 2002; Poppele et al., 2003), suggest that global information about the end position of the legs is coded by single neurons in the spinal cord and spinocerebellar tract. With this in mind and taking into account an assumed redundancy in the locomotor system, I present three new ‘global’ reflex models and show the effect they have on locomotion and postural stability.

I used a computer model of quadruped locomotion, the ‘Robocat’ model, to assess the effects of the introduced reflex- and control-models. Using parametric variations of forces in the actuators of the model, sets of simulations were gathered. To quantify the effect in postural stability for the simulation sets with the different model setups, a new evaluation method was proposed that used a nearest neighbour algorithm.

In this section, the fundamental layout and mechanisms underlying locomotion are presented. Finite state (‘If-Then’) rules for locomotion are described and a brief overview of Fuzzy Logic is presented.

1.1 Movement control

Mammals use three major bodily systems to produce movement: the skeleton, muscles and the nervous system. The skeleton acts as the supporting frame for the muscles and other organs. The muscles, in turn, actuate the movement by pulling on the tendons attached to the skeletal structure. For most joints there is at least one flexor muscle operating the joint and one extensor muscle (Kandel et al., 2000). The nervous system controls the muscles through motoneurons that activate muscle fibres grouped together as motor units (Kandel et al., 2000).

The motor system is the neuromuscular system controlling movement. The neural component is hierarchically organized with the peripheral nerves and spinal cord at the lowest level. The spinal cord contains neural circuits that mediate a variety of reflexes and rhythmical patterns. The reflexes and rhythmical patterns can be modified by higher centres in the central nervous system. This control is mediated through axons projecting from the supraspinal centres to interneurons or directly to the motoneurons. (Kandel et al., 2000)

The next level in the hierarchy of the motor system is the brainstem. In the brainstem, there exist two systems, the medial descending system controlling posture and the lateral descending system controlling distal muscles. These systems receive projections from the cerebral cortex and project down to the spinal cord. (Kandel et al., 2000)
At the highest level are the premotor and primary motor areas of the cerebral cortex. The premotor areas are mainly involved in planning motor action and have projections to both the primary motor areas and the spinal cord (Purves et al., 2001). A variety of reflex circuits in the brainstem and spinal cord simplify the coding of movements in the cortex (Kandel et al., 2000). By exciting some circuits and inhibiting others the wanted movement can be achieved, with the lower areas handling the timing through sensory feedback. In rhythmic movements such as locomotion, alternating activation of antagonistic muscles groups is already coded in the reflexes in the spinal cord, thus descending signals need not be precisely timed. (Kandel et al., 2000)

Another way of viewing the hierarchical structure of the control mechanism of locomotion is presented by Orlovsky et al. (1999). This is a more generalized layout of the control that applies for a variety of species including crayfish, cats and humans, see Fig. 1. Despite the enormous diversity in the structure of the locomotor organs and the central nervous system of different species, locomotor control is based on a few fundamental principles (Orlovsky et al., 1999). The activity of the locomotor organs (trunk, wings, legs, etc.) generates a propulsive force that leads to locomotion. Each locomotor organ is rhythmically innervated by a special neural control network generating the rhythm. This is true for most locomotion as for example stepping and swimming. These neural control networks interact so as to coordinate the locomotor action and they also adopt their activity to the surrounding environment through sensory feedback. The higher hierarchical levels will not be described here, but can be found in further readings (Orlovsky et al., 1999).
Fig. 1. A generalization of the hierarchical structure of locomotion as described by Orolovsky et al. (1999).
1.2 Spinal reflexes

In this section, the basic physiology of spinal reflexes and some of the models that have been developed are presented. There are two main inputs to the spinal cord: the length sensing muscle spindles and the load sensing Golgi tendon organs. This information contributes to the activation of the muscles through several known or hypothesized neural mechanisms, see Fig. 2, and thus the force in the muscle.

In the 1800’s and 1900’s, Charles Bell and Charles Sherrington stressed the importance of sensory feedback in the control of movement. Sherrington popularized Sechenov’s earlier proposal that simple reflexes, taking information from somatosensory receptors, are the basic units of movement and that more complex movements are simply combinations of reflexes. This view does to some degree still prevail, but scientists now believe that reflexes are integrated with centrally generated motor commands, thus producing adaptive movements (Kandel et al., 2000).

Stretch reflexes are mediated by networks of neurons, organized in the form of feedback loops, from somatosensors to motoneurons. They can be monosynaptic, the somatosensory neuron directly projecting to the motoneuron, or polysynaptic, with one or several interneurons in between the somatosensory input and the motoneuron. The reflexes have three important features. First, the reflex pathways can be modified according to the motor task at hand (Prochazka, 1996b; Orlovsky et al., 1999; Kandel et al., 2000). Second, a local sensory input can elicit a response in many muscles (Kandel et al., 2000). Third, supraspinal centres may modulate the reflex, and even in some cases reverse its response (Grillner, 1975; Conway et al., 1987; Orlovsky et al., 1999).

One of the most studied reflexes is the stretch reflex. This reflex contracts a muscle if the muscle is being stretched (Kandel et al., 2000). This is a monosynaptic reflex taking input from muscle spindles\(^2\) relaying it through negative feedback to the motoneuron controlling the muscle. The stretch reflex also has a polysynaptic pathway, Ia-afferents projecting to Ia-inhibitory interneurons, which in turn inhibit motoneurons of the antagonist. Thus, the stretch reflex coordinates the muscles surrounding the joint, activating the stretched muscle while relaxing the antagonist (Kandel et al., 2000). The stretch reflex thus provides the principal mechanism by which voluntary movements as well as more complex reflexes link together contractions of muscles.

Though there are some variations between species, the basic spindle characteristics, shown in acute experiments, are remarkably similar in cats, monkeys and humans (Prochazka, 1996b). The muscle spindles are imbedded in the muscle. They are built up of 6-10 intrafusal muscle fibres, with 1-2 primary, Ia, and 1-5 secondary, II, sensory nerve endings around the noncontractile portion of the muscle fibres. The noncontractile part with the nerve endings is encapsulated which gives the spindle its fusiform shape. The end poles of the intrafusal muscle fibres are attached to the surrounding extrafusal muscle fibres. \(\gamma\)-fusimotor axons and \(\beta\)-skeletofusimotor axons innervate the intrafusal muscles and thus regulate the afferent bursts (Prochazka, 1996b).

---

\(^1\) Somatosensory inputs are the nerve impulses from both muscles and skin receptors.

\(^2\) A muscle spindle is an sensory organ imbedded in the muscle. Its main function is to signal changes in the length of the muscle, thus giving information about the angle of the joint the muscle is controlling.
The muscle spindles show various nonlinearities. By controlling the γ activation both the response to length as well as to velocity can be altered (Prochazka, 1996b). There are several linear models for the muscle spindle, dependent on length variations of the muscle with respect to rest length (Prochazka, 1996b). However, a limitation of the linear models is that the velocity-sensitivity component of the muscle spindle Ia-afferents does not scale linearly with velocity. A power-law relationship between velocity (v), length (l), constant (c) and the firing rate of Ia-afferents has instead been proposed (Prochazka, 1996b):

\[ I_a \text{ firing rate} = c \times l \times v^{0.3}, \]  

(1)

A. Prochazka (1996b) concludes in these four points the characteristics of muscle spindles in voluntary movements in cats’ hindlimbs:

1. Ensembles of 4-5 afferents can provide CNS with accurate high-resolution information on muscle length and force.
2. The linear model, when used on the average length signal, corresponds to the ensemble firing rates. To a first approximation, the spindle afferents therefore signal muscle length and velocity during the step cycle.
3. Deviations from the model suggest some linked action. In other words activation of muscles coactivates intrafusal muscles thus augmenting the response from the spindle or at least maintaining the firing rate as the muscle shortens.
4. The peak Ia inputs from a single muscle to the CNS during the locomotor step cycle is 20-40 k impulse/s. This is nearly enough to depolarize low-threshold alpha-motoneurons of that muscle.

Another reflex pathway is the Ib-inhibitory pathway. It takes information of the load on the muscle from the **Golgi tendon organs**. Via Ib-afferent fibres, this information is transmitted to the Ib-inhibitory interneuron that in turn inhibits the motoneuron. This pathway was first thought to have a protective function, only firing when the tension was high, but experiments have shown that they also signal small changes (Prochazka, 1996b; Kandel et al., 2000). Hence, the Ib-inhibitory pathway mediates negative force feedback.

Feedback control studies on tendon organs have shown that they exhibit a high-pass filtering property, (Kandel et al., 2000), thus the dynamic component of the force signal is passed on more easily than the static. In other words Ib-afferents fire more when the load changes than for the static load. The transfer function for the load-Ib afferent system can be expressed as:

\[ I_b(s) = K \frac{(s + 0.15) \cdot (s + 1.5) \cdot (s + 16)}{(s + 0.20) \cdot (s + 2.0) \cdot (s + 37)}, \]  

(2)

where the K is a gain constant (Prochazka, 1996b). However, on top of this tendon organs also have some nonlinear characteristics. First, the fibres going in to the receptor capsule might be unloaded because the muscle fibres not connected to the capsule are contracted while the ones connected to the capsule are not. Secondly, the relationship between the load

---

3 The Golgi tendon organ is a sensory receptor located at the junction of the muscle fibres and the tendon. Because the Ib-afferent ending is intertwined with the collagenous fascicles of the tendon, small changes in muscle force are sensed.
on the whole muscle and the load on the muscle fibres entering the receptor capsule is non-linear. Finally, the recruitment of each motor unit gives a step in firing rate. (Prochazka, 1996b)

As stated above, the sensitivity (gain) of reflexes varies. Studies have shown that rhythmical presynaptic inhibition of Ia-afferent fibres modulates the gain of transmission in the monosynaptic stretch reflex during locomotion (Conway et al., 1987). Thus, the strength of the stretch reflex is altered during different locomotor states. The Ib-inhibitory pathway also altered. It goes from being inhibitory during static stance to excitatory during locomotion by suppression of the inhibitory pathway and facilitation of the excitatory pathway (Conway et al., 1987). This phenomenon is called state-dependent reflex reversal (Kandel et al., 2000).

1.3 Locomotion in Cats

Much of the research on mammalian locomotion has focused on experiments in the cat. However, primitive vertebrates such as lampreys and invertebrates such as cockroaches have also given important insights into the control of locomotion (Orlovsky et al., 1999; Kandel et al., 2000). There are many different forms of locomotion, swimming, walking, flying, etc. All locomotion has a common feature, namely the rhythmical movement of the body and its appendages (Orlovsky et al., 1999; Kandel et al., 2000). Another common feature is the modification through sensory input of the rhythmical locomotor pattern to better suit the environment (Kandel et al., 2000). These features have been well studied in experiments with cats on treadmills. I shall mainly concentrate on research gathered from such experiments.

As in other walking animals the locomotor step cycle in the cat consists of two principal parts, stance (support) and swing (transfer) phase (Grillner, 1975; Orlovsky et al., 1999; Kandel et al., 2000). The swing phase starts when the leg is in its most posterior position in relation to the body. The leg is lifted off ground and is moved forward until it reaches the extreme anterior position. When in this position the paw makes contact with ground and the stance phase starts. During this phase, the leg is loaded by part of the body weight and, as the leg moves backwards towards the extreme posterior position, it also develops a propulsive force. (Orlovsky et al., 1999)

There are three major joints involved in the step cycle: hip-, knee-, ankle-joint. They all perform considerable flexion and extension throughout the cycle. The temporal flexion-extension pattern of the hip joint is the simplest, with one peak in flexion when the leg is in anterior extreme position and one peak in extension determining the posterior extreme position. The patterns of the other joints are more complex with two peaks in flexion and two peaks in extension (Grillner, 1975; Orlovsky et al., 1999). The extra peaks are necessary to shorten the total length of the leg in mid-Stance in order to reduce vertical oscillations and to retract and lift the foot to avoid tripping in the swing phase (Orlovsky et al., 1999).

The temporal characteristic of the step cycle is highly speed-dependent. As speed increase, the cycle duration shortens proportionally. The stance phase is the main contributor to the shortened duration, while the duration of the swing phase remains nearly constant with varying speeds (Grillner, 1975). The same phenomenon is also observed in crustacean and insects, suggesting that different species may have the same functional organization in their leg controllers (Orlovsky et al., 1999).
1.3.1 The CPG and some rhythm-generator models

During walking, a neural network that has been called the Central Pattern Generator activates the motoneurons of leg muscles, creating rhythmical movements of the legs. It is constructed of a rhythm-generator and a patterning network, the output of which results in the complex activation pattern of the motoneurons.

The neural rhythm for locomotion in cats, and also in other animals including humans, has been shown to arise from local networks in the spinal cord (Kandel et al., 2000). This local network, originally called the Intrinsic factor by T. Graham Brown (1911) is called the Central Pattern Generator, CPG. The CPG creates the motor pattern for locomotion. The existence of a CPG has been verified in spinal preparations of cats on treadmills, showing that neither the cerebral cortex nor the brainstem are essential to produce the locomotor patterns (Kandel et al., 2000). Even spinal cats in which all sensory input to the spinal cord below the transection has been removed can carry out rudimentary locomotion with pharmacological facilitation (Prochazka, 1996b).

Though the general pattern of motoneuron excitation follows the two phases mentioned above of flexion and extension, some motoneurons have several peaks of activity, and in others the burst of activation is phase shifted with regard to the general flexor-extensor pattern. This suggests that there is some sort of transformational network for some motoneurons (Grillner, 1975; Orlovsky et al., 1999). Therefore the CPG can be viewed as a neural network consisting of an oscillatory rhythm generator and a patterning network moulding this biphasic signal into a more complex locomotor pattern (Grillner, 1975; Kandel et al., 2000).

The oscillatory network comprising the rhythm generators in mammals has yet to be identified due to the great complexity of the mammalian spinal cord (Kandel et al., 2000). Several hypotheses have been presented and I will briefly describe two, the biphasic and the triphasic.

T. Graham Brown showed that the isolated spinal cord can produce a rhythmical pattern without sensory feedback. He proposed a biphasic arrangement of two groups of neurons inhibiting each other and named them half-centres. (Grillner, 1975; Orlovsky et al., 1999; Kandel et al., 2000) Later in the 1960s, this hypothesis was supported in experiments using the drug L-DOPA. The interneurons for flexion were found to inhibit the interneurons for extension, and vice versa, thus producing alternating bursts of extension and flexion (Kandel et al., 2000).

---

Fig. 3 Tri-phasic generator. The rhombuses signify excitatory inputs and dots are inhibitory. a, the original tri-phasic generator. b, Gelfand’s model incorporating both biphasic and tri-phasic characteristics. In this setup, the stepping pattern is mediated by the tonic excitation of the E group. S is sensory input, inhibitory at the extreme extension position of the joint else excitatory.

---

4 For an overview of different animal preparations see Appendix I.
The tri-phasic generator was originally proposed to explain the respiratory rhythm generation. It has since been shown to explain the scratching rhythm well, which in many ways is similar to the stepping rhythm. It consists of three groups of interneurons, see Fig. 3 a. Group 1, corresponds to flexor interneurons, group 2 to timing interneurons and group 3 to extensor interneurons. Group 3 interneurons work to switch activity from flexor phase to extensor phase. While group 1 is active, the activity in group 2 interneurons gradually increases and when the activation reaches a predefined level group 2 excites group 3 interneurons. Some group 3 interneurons inhibit group 1 interneurons, thus inhibiting flexion and at the same time depriving group 2 of some of its excitation. (Orlovsky et al., 1999)

The half centre model predicts a simple biphasic distribution of activity of locomotor-related interneurons in the spinal cord. However some studies have pointed to a more complex phase-related distribution of interneuronal activity during stepping. The findings suggest that interneurons with the same functionality as the group 2 interneurons in scratching are present in the generation of the locomotor rhythm (Orlovsky et al., 1999).

In an attempt to incorporate the findings for both biphasic and three-phasic generators, Gelfand et al. in 1988 constructed a mix of the two models (Orlovsky et al., 1999). This network can produce both scratching and stepping patterns through a tonic excitation of either the \( F_1 \) group creating a scratching pattern or excitation of the \( E \) group creating a stepping pattern. The inputs from mechanoreceptors control the switching between phases, preventing premature switches (see Fig. 3 b). (Orlovsky et al., 1999)

### 1.3.2 The motor pattern

During gait, a pattern of the muscle contractions arise: the locomotor pattern. This pattern has been subdivided into functional phases as outlined below.

The rhythmical pattern of contractions in the muscles during gait is called the locomotor pattern. Many muscles are involved and the timing of the contractions follows a complicated scheme. The pattern has been divided into two main phases, the stance and the swing phase (Grillner, 1975; Prochazka, 1993; Orlovsky et al., 1999; Kandel et al., 2000). The same phases have been identified in many different animals and have even successfully been used in above-knee prosthesis in humans (Prochazka, 1993). The phases can further be divided into four sub-phases, \( F \) (flexion), \( E_1 \) (extension), \( E_2, E_3 \) (Grillner, 1975; Orlovsky et al., 1999; Kandel et al., 2000). \( F \) and \( E_1 \) are sub-phases of the swing phase while \( E_2 \) and \( E_3 \) are sub-phases of the stance phase, see Fig. 4.

The flexor muscles in ankle, knee and hip joints become active just before the swing phase onset (Orlovsky et al., 1999). The swing phase thus begins with the flexion sub phase (\( F \)), where the knee, hip and ankle joints are flexed (Kandel et al., 2000). The contractions in the flexors of the hip joint bring the leg forward in relation to the body. The contraction in the knee and ankle flexors shortens the length of the leg, thus raising it from the ground during the first half of the swing phase (Orlovsky et al., 1999). Later in the \( E_1 \) sub-phase of Swing, the flexor activity stops and extensors become active, extending the leg until it touches the ground (Orlovsky et al., 1999). This is done in anticipation of supporting the weight later in the stance phase (Kandel et al., 2000).

The extensor muscles have a very similar activation pattern. Knee and ankle extensors become active at the end of the swing phase (\( E_1 \)), in anticipation of foot contact, and stay active throughout the stance phase (\( E_2-E_3 \)). This creates a gross extensor synergy, making the limb
rigid enough to bear its portion of body weight (Orlovsky et al., 1999). In the E₂ sub-phase of early Stance, large extensor muscles are excited, yet the knee and hip joints continue to flex due to yield in the extensor muscles (Kandel et al., 2000). This yielding is essential for an efficient gait since it smooths out the vertical oscillations (Orlovsky et al., 1999; Kandel et al., 2000). As the cycle continues into the E₃ sub-phase the joints become more extended, thus increasing the length of the leg. This increase in length of the leg at the end of the stance phase provides a propulsive force to the body as well as an antigravitational force (Orlovsky et al., 1999).

This basic pattern of activity occurs at all locomotor speeds. A number of experiments suggest that stepping at different speeds and intensities is controlled by one neural network and that the differences in speed are the result of changing certain inputs to this network. (Orlovsky et al., 1999)

Significant deviations from the general pattern just described can occur when there is strong phasic drive in descending pathways, strong afferent input, or drug effects (Orlovsky et al., 1999; Kandel et al., 2000). The models described in this paper do not include phasic descending influences or drug effects, but afferent contributions are key aspects of the study.

1.3.3 Sensory input

Sensory input controlling the CPG is important for the stabilization of locomotion. Perturbations can disrupt the coordination of biomechanical events with the CPG-activation pattern, which can lead to a fall. Three of the modalities of sensory input and their impact on the CPG to stabilize locomotion are presented below.

For locomotion to be efficient the basic motor pattern must be adapted to suit the environment. In daily life, vestibular information on head tilt and anticipatory information from the visual cortex are important for the avoidance of obstacles and stabilization of gait. However, in this study I limited the model to the spinal network.

The locomotor movements must adapt to the environment by reacting to and compensating for various types of perturbation (Grillner, 1975). Through reflexes, somatosensory input shapes the locomotor pattern appropriately (Ryback et al., 2002). One view is that the reflexes controlling the CPG come into play only when there are deviations from the sensory input anticipated by the CPG (Grillner, 1975; Orlovsky et al., 1999; Kandel et al., 2000). I shall now introduce three important reflexes involving obstacle avoidance, hip extension, and force feedback from Golgi tendon organs.

As stepping rate increases the stance phase portion of the step cycle is reduced whereas the swing phase duration remains relatively constant, (Grillner, 1975; Orlovsky et al., 1999; Kandel et al., 2000). Several experiments with cats and dogs have indicated that the switching between the stance phase and the swing phase is triggered by the angle between the hip and the trunk (Kandel et al., 2000). A good estimate of the angle of a joint can be derived from the activity of an ensemble of muscle spindles in muscles being stretched, see 1.2 Spinal reflexes. In this particular case, the burst activity from muscle spindles in the hip flexor muscle is used. These reflexly facilitate flexor motoneurons and inhibit extensor motoneurons, and thus contribute to the neural decision to switch from Stance to Swing (Kandel et al., 2000). It has been found that stopping a leg in E₁ will result in continued extension until the hip joint angle is in a position corresponding to E₃, then flexion is initiated. All the while, the contralateral leg continues its stepping motion. (Grillner, 1975) The initiation of the flexion phase is also found to be dependent on where in the step cycle the contralateral leg is, which serves to coordinate the limbs for in-phase and counter-phase locomotion (Orlovsky et al., 1999).

Experimental results from many different species show that one of the critical points in the step cycle is the transition form Stance to Swing (Orlovsky et al., 1999). This is where the leg's support function is terminated. If at the end of the stance phase, the leg is still loaded, initiation of Swing would tend to destabilize the animal. To avoid this, the swing phase is not
initiated until the leg is unloaded (Orlovsky et al., 1999; Kandel et al., 2000). Golgi tendon organs in extensor muscles signal extensor load and in the stance phase their afferents excite extensor motoneurons (Kandel et al., 2000). Note that the sign of this reflex is the opposite to that of the Ib-inhibitory reflex action that occurs during static posture see 1.2 Spinal reflexes.

Thus, the contribution of the Golgi tendon organs is state-dependent: it is equivalent to negative force feedback in static postures, and positive force feedback during locomotion. Both the afferent information from the muscle spindles and the Golgi tendon organs are important in weight bearing. In cats more than 50% of the excitatory input to the extensor muscles during static stance is lost if the Ia- (muscles spindles) and the Ib- (Golgi tendon organs) afferent input is removed (Kandel et al., 2000). It was suggested by Prochazka et al. (2002) that the stretch reflex response was too weak and too delayed to contribute significantly to weight bearing during walking. However, through computer simulations, it has now been shown that the stretch reflex response helps rescue locomotion from instability when the central CPG activation is too low, but it can destabilize locomotion when CPG activation is adequate (Yakovenko et al., 2004).

The greater impact of the leg’s movements on the step cycle, and ultimately on the animal’s whole posture, during the stance phase demands more afferent feedback than in the swing phase. In the swing phase the Ia inhibitory effect on the antagonist is reduced due to a phasic gamma modulation (Orlovsky et al., 1999). As stated above, the fact that the swing phase temporal characteristic is more or less the same for varying speeds of gait, and the fact that the E1 phase starts before foot contact with the ground both suggest that the swing phase is less dependent on afferent information than the stance phase (Orlovsky et al., 1999). However, input from skin receptors can under some circumstances strongly influence the CPG. Especially, studied is the stumbling reaction in cats. In swing phase, a small mechanical stimulus applied to the dorsal part of the paw elicits a reflex that promotes flexion of the knee- and ankle-joints and extension of the contralateral leg, moving the paw away from the stimulus (Orlovsky et al., 1999). This reflex thus results in an over-stepping of the obstacle, thus preventing the cat from stumbling. As with the Golgi tendon organs, this reflex is phase dependent. In stance phase the same stimulus to the dorsal part of the paw elicits the contrary response, in other words an excitation of extensor muscles reinforcing the ongoing extensor activity in the stance phase (Kandel et al., 2000).

### 1.4 If-Then rules

As was stated in the previous section on Locomotion in Cats, sensory inputs affect how the CPG controls the step cycle. Three types of afferent input have been shown to be particularly influential: Ia-input from muscle spindles, Ib-input from Golgi tendon organs and input from skin receptors. In the following section, I will discuss some ideas derived from technology that have helped formulate and test hypotheses concerning the interaction between sensory input and CPGs.

In above-knee prostheses, various forms of finite-state control have been implemented since the late 1960’s (Prochazka, 1993). As described in 1.2 Spinal reflexes, some of the responses elicited by mechanical stimuli during locomotion in cats are phase dependent. This phase dependent reversal of reflexes may be viewed as a special case of finite-state control in the form of IF-THEN rules (Prochazka, 1996b). Interestingly, similar rules controlling locomotion have been found independently in cat, cockroaches, lobsters and stick insects (Grillner, 1975; Prochazka, 1996b).
Popović (1993) showed that a finite-state control model could be used in restoring locomotion in paralyzed humans. In his paper an invariant pattern of outputs in relation to states were used. The output criteria were in the form of a set of IF-THEN rules connected to the each state. In a finite state model the following elements are needed:

\[
S = \{s(j), j=0, 1, \ldots, i\}
\]  \hspace{1cm} (3)

\[
s(j) = \delta_j[s(j-1), X(j)]
\]  \hspace{1cm} (4)

\[
X(j) = \{x(1,j), x(2,j), \ldots, x(k,j)\}
\]  \hspace{1cm} (5)

\[
Y(j) = \{y(1,j), y(2,j), \ldots, y(n,j)\}
\]  \hspace{1cm} (6)

where \(k\) is the number of inputs, \(n\) the number of outputs and \(i\) the number of states. The \(\delta_j\) function is a mapping from the inputs and the previous state to the new state, and this mapping is invariant (Popovic, 1993).

The Robocat model, developed by Yakovenko et al. (2004), uses a form of finite state space control based on two explicit IF-THEN rules and a time-out rule for the continuous clock controlling the CPG-activation patterns. The CPG-activation pattern is divided into stance and swing phases and the If-Then rules, when triggered, move the readout position, \(t_i\), for the CPG-pattern in between the phases. These rules thus function as the \(\delta_j\) functions for the CPG, jumping over certain parts of the activation pattern. The rules that were used in the original model were:

1. **Stance → Swing**: IF readout in stance phase AND ipsilateral hip is extended AND ipsilateral leg is unloaded, THEN move readout for the CPG to the swing phase.
2. **Swing → Stance**: IF readout in swing phase AND ipsilateral hip flexed AND ipsilateral knee is extended, THEN move readout of CPG to the stance phase.

(Yakovenko et al., 2004)

Another rule base was proposed by Prochazka et al. (2002) which takes into account interlimb coordination and more elaborate conditions such as tripping over objects or losing ground support in one limb.

**Stance phase**

- **Stance-Swing**: IF Stance AND extensor force low AND hip angle large AND contralateral limb supported, THEN switch to Swing
- **Foot-in-hole**: IF mid-Stance AND no ground contact AND contralateral limb supported, THEN switch to placing reaction

**Swing phase**

- **Swing-Stance**: IF Swing AND hip angle small AND knee angle large THEN switch to Stance
- **Tripping**: IF Swing AND skin stimulus to font of foot THEN switch to placing; IF Stance AND skin stimulus to front of foot THEN prolong Stance

There are several ways of constructing a controller based on these rule bases. In the original Robocat model the rules were Boolean. All inputs to each rule had their own specific threshold. If and only if all inputs exceeded their thresholds, the rule would fire, and the CPG-readout position would be moved. As these rule bases are based on specialist linguistic interpretations of events in the locomotor cycle, another way of implementing the model control system would be in the form of a Fuzzy Logic Controller (FLC). The main advantage of FLCs is their ability to control complex systems based on linguistic interpretations of the control
actions required, as determined by an expert human operator, without the need for mathematical models of the system being controlled.

1.5 Fuzzy Logic control

Fuzzy Logic control, FLC, is a control structure built on linguistic rules instead of a model of the system. Such a rule is for example, IF water level is rising little THEN open valve little. This form of linguistic reasoning has the advantage that the rules are easily interpreted and that expert operator knowledge can be used directly on the system. To interpret these rules, FLC uses fuzzy set theory and fuzzy logic. This was formalized by Lotfi Zadeh during the 1960s (Cox, 1994). Zadeh proposed this form of linguistic reasoning to handle, as he states, the problem that our ability to make precise and significant statements diminishes as complexity of a system increases (Cox, 1994). Since this form of logic and set theory differs quite drastically from the normal crisp logic and set theory, I shall start by giving a brief introduction to fuzziness, fuzzy set theory and fuzzy logic.

1.5.1 Fuzziness

According to Zadeh fuzziness is the description of the intrinsic imprecision in our intuitive judgment of categorized membership (Wang, 1996). When referring to a property of an item it is difficult to define a strict interval in within which the property is defined. In the example: The man is TALL, we refer to the domain of discourse height. Given that the man is 1.75m, would he still be considered tall if he was only 1.70m? Since the property tall changes continuously, we cannot determine a specific height when the man goes from being tall to not tall, see Fig. 5. As can be seen from this example our judgement of a property has an intrinsic imprecision.

According to Cox (1994), the phenomena we encounter in everyday life are imprecise. They hold a certain degree of fuzziness in their nature. Looking at the domain of discourse, the height of a man, we can easily divide the positive real axis into at least three regions with the labels: small, normal, tall. The crisp subdivision of the real axis would give a contra intuitive result when at the borders of the regions. Take for example a man that is of normal height when he goes to sleep and when he wakes up, one millimetre longer, he is suddenly tall. Our linguistic labels have an intrinsic smoothness that is not represented with crisp regions.

Looking at men of varying height, we would say that 1.65m does not conform to the label tall but as their height gradually increases, they conform more and more to the label tall. From, let us say 1.75m, the men fully belong to the label tall. On the other hand, men would comply fully with the label normal at a height of 1.65m and as their height increases they progressively lose their membership in that label, but gain in membership in the label tall. Fuzziness is this concept of intertwined and graded memberships of linguistic labels/sets.

Zadeh introduced membership functions to map the fuzziness in our linguistic labels to actual numbers (Wang, 1996). The membership function scores the fractional level of truth of statements in the following way: “The membership of John’s height (1.74m) in the category tall is 0.9” Zadeh means that this should not be interpreted as a probability. John’s height is
not a random number that in 90% of cases takes the value tall, but rather 0.9 is the degree to which the label tall and 1.74m are compatible (Wang, 1996). According to Wang (1996), Zadeh understands membership functions as having both subjective and context-dependent attributes, and thus no general method can be adopted to define them. Furthermore, the concept of grades of membership or fuzziness has not yet been fully interpreted (Wang, 1996). As the membership functions greatly influence the properties of FLCs, (Cox, 1994), this presents problems for their designers. Nevertheless, a large variety of devices controlled by FLCs have been produced and different forms of gradient searching techniques have been implemented to improve the performance of FLCs (Lee, 1990a; Chen et al., 1997; Davoodi and Andrews, 1999).

### 1.5.2 Fuzzy sets

All sets are definitions of membership in a universe of discourse, $U$, see Fig. 6. Crisp sets are Boolean in nature, i.e. the point in $U$ either belongs to the set or not. The membership function for the crisp set, $F$, thus maps a point in $U$ to $\{0,1\}$, or $\mu_F: U \rightarrow \{0,1\}$. As described earlier, these crisp sets are too strict to describe our intuitive linguistic sets. Fuzzy sets may be viewed as a generalization of the concept of crisp ordinary sets, by letting the membership function map $U$ to $[0,1]$, $\mu_F: U \rightarrow [0,1]$ (Lee, 1990a).

Thus, any and all points in $U$ are members of the set to a certain degree. By defining the membership as a degree of membership, the fuzziness of the linguistic sets can be modelled by fuzzy sets and thus enabling the fuzzy set theory to handle linguistic statements.

### 1.5.3 Membership functions

The membership function is the function, $\mu_F$, which maps the point in the discourse space to the truth of that point belonging to $F$. Since these functions, as stated above, are subjective and context dependent, they may assume almost any form. The most common form is the triangular or the trapezoid, see Fig. 5 a,b, but also Gaussian bell curves and PI-curves among others may be used (Driankov et al., 1993; Cox, 1994; Castro, 1995). Although the shape of and the subdivision of the discourse space with membership functions is critical for the performance of FLCs, experiments point to a high degree of tolerance (Lee, 1990a, b; Driankov et al., 1993; Castro, 1995).

### 1.5.4 Fuzzy Logic

Fuzzy Logic is much closer in spirit to human thinking and natural language than the traditional logical systems (Lee, 1990a). It is an approximate reasoning system making use of fuzzy sets to infer conclusions. Fuzzy Logic, like classical logic, uses two forms of logical implications, Modus Ponens and Modus Tollens, to infer the existence of a consequent state.

Beginning with classical logic, a statement is given: $P \Rightarrow Q$ where $P$ is the premise that implies the consequent $Q$. If we know that the premise $P$ is fulfilled, then we can safely conclude the consequent $Q$. This is the Modus Ponens of the classical logic. The Modus Tollens is the backward reasoning, given the statement: $P \Rightarrow Q$, and the knowledge that $\neg Q$ ($Q$ does not exist), the conclusion that $\neg P$ ($P$ does not exist) can be drawn. (Lee, 1990b; Cox, 1994)

With the use of fuzzy sets, the truth of the existence of $P$ and $Q$ is not Boolean as in classical logic. Thus, the answer to a Modulus Ponens or Tollens is widened to include all degrees
in-between the Boolean answers *true* and *false*. In order for Fuzzy Logic to infer a consequence for the non-Boolean P and Q, Fuzzy Logic uses the membership functions in its inference system. The statements are given as: \( \mu_P \supset \mu_Q \) and can be interpreted as “a membership in P implies a membership in Q”. Hence, Modulus Ponens reverts back to the truth of \( x \) belonging to P, \( \mu_P(x) \), implies that we can infer the membership function for Q, \( \mu_Q(y) \). In the same way, Modulus Tollens becomes: the truth of \( y \) not belonging to Q, \( \neg \mu_Q(y) \), implies that we can infer the membership function of \( \neg P, \neg \mu_P(x) \). (Lee, 1990a, b; Cox, 1994)

The use of fuzzy sets in Fuzzy Logic gives rise to ambiguities not found in classical logic, for example a statement can be both true and not true at the same point. Therefore, new form of inference and implication is needed. Implication and inference can be modelled in different ways, but the most common are the Mamdani and the Takagi-Sugeno inference systems. The Mamdani implication and inference system was introduced in 1975 (Mamdani and Assilian, 1975). It uses a min-max system described below.

For two rules on the form:
\[ R_1: \text{if } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } z \text{ is } C_1 \]
\[ R_2: \text{if } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } z \text{ is } C_2 \]
where A, B and C are membership functions in different domains of discourses, the Mamdani use the following three steps to infer the output membership function:

1. \( R_i: \alpha_i = \mu_A(x) \land \mu_B(y) \)
2. \( \mu_{C_1}(w) = \alpha_i \land \mu_C(w) \)
3. \( \mu_{C_2}(w) = \mu_{C_1}(w) \lor \mu_{C_2}(w) = [\alpha_i \land \mu_{C_1}(w)] \lor [\alpha_i \land \mu_{C_2}(w)] \)

where the \( \land \) symbol meaning minimum, and \( \lor \) symbol stands for maximum. This form of reasoning is very intuitive and that is its main advantage as compared to the Sugeno system. The Takagi-Sugeno system uses another set of rules and it gives a crisp output instead of a membership function. The rules are of the form:
\[ R_i: \text{if } (x \text{ is } A_i \ldots \text{ and } y \text{ is } B_i) \text{ then } z = f_i(x, \ldots, y) \]

The inference is done by a weighing of the output functions:
\[
z_0 = \frac{\sum \alpha_i f_i(x_0, \ldots, y_0)}{\sum \alpha_i},
\]

1.5.5 The control system

The Fuzzy Logic controller, FLC, is built up of four major parts: the fuzzification, the inference system, the defuzzification and the database, see Fig. 7. In order for the inference system to be able to determine the right control, the crisp input values must be converted into membership functions. This conversion is made by the fuzzifica-
tion, where the crisp values are mapped to the membership representation. Before the actual conversion into membership functions, it might in some cases be favourable to first transform the input. If so the transformation of the inputs must entail an output transformation after the defuzzification part. (Cox, 1994) Dependent on whether the inference system is based on composition or on individual-rule-firing based the conversion to membership functions can only be performed in two ways, (Driankov et al., 1993).

The inference engine is the part of the FLC that takes the membership of the inputs and combines it with the rule base to deduce the correct output. As stated above, there are several different forms of inference systems but as the Mamdani min-max rule of inference is the inference system used in the experiment below, I will only describe this system. The inference engine can be composition-based or individual-rule-firing based. The composition-based inference engine lets all rules fire at the same time while the individual-rule-firing based inference engine evaluates one rule at a time and after that evaluates the output memberships (Driankov et al., 1993). The most common type is the latter (Driankov et al., 1993). However, for the Mamdani min-max inference system the choice is irrelevant since the two approaches yield the same result (Driankov et al., 1993).

The database is the component holding the information about the membership functions and the rule base. When choosing the type of representation of the linguistic terms the memory efficiency of the membership functions is of great importance. The most popular membership functions with respect to this are the triangular-, trapezoidal-, and the bell-shaped functions. These three choices can be explained by the ease with which a parametric, functional description of the membership function can be obtained. They can be stored with minimal use of memory, and manipulated efficiently by the inference engine. (Driankov et al., 1993)

Having chosen the membership representation, the mapping of the linguistic memberships on the domain of discourse can greatly affect the performance of the FLC. For the input membership functions there are two parameters that are important, the cross-point level and the overlapping of the membership functions. The cross-point level is the degree of membership at the cross-point of two crossing membership functions, see Fig. 8 a. If all values of the input domain have a cross-point level strictly greater than zero, this would entail that every crisp input belong to a membership function. If this is not the case, some crisp input values will not be matched to rules and thus no control output can be inferred to that point. The normal value of the cross-point level is 0.5 and it has been shown for linear systems up to the third order to be the optimal level. (Driankov et al., 1993)

If two membership functions have a cross-point level greater than zero, they will overlap the input domain in one area. If this area spans the distance between the input membership functions peak-value points, then the output will change smoothly using Centre-of-Gravity defuzzification. If there is a set of input values that only have a membership in one membership function the output will be discontinuous, as depicted in Fig. 8 b,c. (Driankov et al., 1993)

Fig. 8. a shows the definition of cross-points, as the point in the domain where the membership functions have the same truth value, and the cross-point level, as the truth value at the cross-point. b, c shows the effect of membership overlap.
For the output membership functions, symmetry is important, i.e. the output membership functions should be symmetric. This has to do with the fact that the defuzzification and the inference system should yield an intuitive output. Given that the FLC uses Centre-of-Gravity defuzzification and non-symmetric output membership functions, the peak of the output membership function and the Centre-of-Gravity will not correspond to the same point (Driankov et al., 1993). Since this makes the control unintuitive, the use of symmetric output membership functions is recommended.

The defuzzification component of the FLC is responsible for converting the output memberships into a crisp control output. There are a number of plausible defuzzification methods, but here I will only describe three of these, namely Centre-of-Area/Gravity, Centre-of-Sums and Mean-of-Max (MoM). The defuzzification methods are defined as:

Centre-of-Area, CoA:

\[
    u^* = \frac{\sum_{i=1}^{l} u_i \cdot \max_{k} \mu_{out,i}(u_i)}{\sum_{i=1}^{l} \max_{k} \mu_{out,i}(u_i)},
\]

Centre-of-Sums, CoS:

\[
    u^* = \frac{\sum_{i=1}^{l} u_i \cdot \sum_{n=1}^{k} \mu_{out,i}(u_i)}{\sum_{i=1}^{l} \sum_{n=1}^{k} \mu_{out,i}(u_i)},
\]

Mean-of-Max, MoM:

\[
    hgt(U) = \sup_{u \in U} \mu_k(u), \forall k \\
    u_{max} = \{u \in U | \mu_k(u) = hgt(U)\}, \forall k, \\
    u^* = \frac{\sum_{i=1}^{l} u_{max}(i)}{n}
\]

(Driankov et al., 1993)

where \( U = \{u_1, ..., u_l\} \), \( u^* \) is the crisp output value, \( k \) is the number of output membership functions.

When comparing the defuzzification methods Driankov (1993) takes into account five criteria: continuity, disambiguity, plausibility, computational complexity and weight counting. The output is defined as continuous if a small change in input does not result in a large change in output. The defuzzification method is disambiguous if two output membership functions with the same area result in a single output. For the methods stated above this does not pose a problem and

<table>
<thead>
<tr>
<th></th>
<th>CoA</th>
<th>CoS</th>
<th>MoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuity</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Disambiguity</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Plausibility</td>
<td>no</td>
<td>no</td>
<td>no*</td>
</tr>
<tr>
<td>Comp. Complexity</td>
<td>bad</td>
<td>good</td>
<td>good</td>
</tr>
<tr>
<td>Weight counting</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

**Table 1.** This is a subset of a table showing the properties of different defuzzification methods in (Driankov et al., 1993). CoA is Centre of Area-, CoS is Centre of Sum- and MoM is Mean of Maxima-defuzzification method. (*) no only in the case of scaled inference.
hence they are disambigous. The plausibility criteria is defined as: a method is plausible if the crisp output is approximately in the middle of the support and for that point the degree of membership is high. Computational complexity is merely how much computational power the method needs to deduce the crisp output. Lastly, the weight counting criteria is the ability for the method to take into account the areas of the output membership functions. When more than one rule infers an output membership function, to different degrees, the method meets the criteria if it weighs all the superimposed memberships to drive the output. The three methods described above meet these criteria as stated in Table 1.

1.5.6 Accomplishments with fuzzy control

Ever since Zadeh in the late 1960s, (Jacobs, 1997), and Mamdani in the early 1970s, (Mamdani and Assilian, 1975), introduced Fuzzy Logic it has proven its strengths in many applications. The main strength is the ability to control nonlinear systems that are hard or impossible to model explicitly (Chen et al., 2004). Both in the field of robotics and physiology, FLCs have been utilized in a wide variety of settings, from controlling the behaviour of a biomimetic robotic fish, (Yu et al., 2004), to classification of EMG-patterns, (Chan et al., 2000; Hussein and Granat, 2002), and determining muscle activation patterns for locomotion, (Jonic et al., 1999). However, to the author’s knowledge, Fuzzy Logic does not appear to have been used in combination with a model of a locomoting cat.

Fuzzy Logic’s ability to control nonlinear systems has been proven in functional electrical stimulation (FES). The behaviour of electrically stimulated muscles is both nonlinear and time variant. Parameters often have to be recalibrated for each subject and for every period of use (Chen et al., 2004). In FES applications, FLCs have been shown to improve overall performance as compared to standard PD-controllers in both cycling, (Chen et al., 1997), and locomotion, (Chen et al., 2004). Both systems used a Mamdani min-max inference systems working on the error and the derivative of the error. This gave the developers more degrees of freedom to shape the fuzzy PD-controller as compared to a regular PD-controller, and the FLC was found to outperform the standard PD-controller.

Three machine learning techniques have been evaluated for the control of FES assisted locomotion in paraplegics (Jonic et al., 1999). The train and test data use by the three ML-techniques was prerecorded data from a model of simulated, FES-assisted locomotion in a human. The aim for the controllers was to generate rules to control a FES-device. In this respect, the ANFIS, adaptive-network-based fuzzy inference system, outperformed both an ANN-based controller and an inductive learning algorithm. The ANFIS technique is based on a fuzzy controller that is tuned by an artificial neural net (Hussein and Granat, 2002). ANFIS has been used in several applications and is almost a black box technique for tuning FLCs of the Sugeno type (Jonic et al., 1999; Hussein and Granat, 2002). Although Jonić et al. (1999) concluded that the best generalization was obtained with an inductive learning algorithm, the use of Fuzzy Logic in the field of locomotion remains promising.
2 Method

A computer model was used to simulate the impact on stability of different systems of reflexes and of global control. A computer model of the walking cat is described in the following section, as are the proposed global reflexes and modifications to the original model. In section 2.2, the implementation of an FLC system is described.

Evaluating the model was done by viewing the maximum isometric force of every musculotendon actuator as a parameter, thus giving the model a six dimensional parameter space in which stability could be evaluated. The data points were gathered both by a random search routine, see section 2.3 Parametric testing routine, and by evaluating the stability on predefined points. The first and second principal component (PC) was extracted from the data points generated by the random search routine of the system with central activation and with the stretch reflexes, implemented by Yakovenko et al. (2004). The predefined points were then places on the plane defined by the two PCs. Each data point represents a simulation run with a certain set of parameters, the six maximum isometric forces in the actuators, and a reflex and control system. The measurement of stability is described in section 2.4 Evaluation principle for Results and is a quantification of the volume, in the six dimensional space, within which the model is stable.

2.1 The model

The original model was developed by S. Yakovenko, V. Gritsenko and A. Prochazka (2004). It is a planar locomotor model of two hindlimbs attached to a torso that is supported by a frictionless wheel. Each leg has four rigid-body segments (thigh, shank, foot and toe) that are activated by six musculotendon actuators (Hip-, Knee- and Ankle- flexors and extensors), see Fig. 9. All the joints are modelled as frictionless revolutes. The control system with activation profiles, reflex responses, etc. was implemented in MatLab version 6.5 (The MathWorks, USA) and coupled to the mechanical model that was implemented in Working Model 2D version 5 (Knowledge Revolution, USA).

The parameters of the mechanical model such as dimensions, masses and moment arms of the actuators were chosen to mimic those of a cat, but as S. Yakovenko et al. were seeking a more general model, that would hold over a larger parameter space, the parameters were only set approximately (Yakovenko et al., 2004). To minimize slipping in the original model the following parameters were set: coefficient of restitution = 0.1 and coefficient of Coulomb friction = 10. A Kutta-Merson integrator with variable integrator steps, and integrator error tolerance set to $10^{-6}$ using 5 significant digits was used to calculate the results.
The simulations were all started just before foot contact and with the initial velocity of the trunk and leading leg of 0.7m/s. The force in each muscle was calculated using a linear force-length relationship and a Hill-type force-velocity relationship.

\[ F = F_{\text{max}} \cdot f_{\text{f}} \cdot f_{\text{l}} \cdot a_t + F_{\text{max}}^{\text{passive}} \cdot f_{\text{l}}^{\text{passive}}, \]  
\[ \dot{a}_t + \left( \frac{\tau_{\text{act}}}{\tau_{\text{deact}}} \right) \cdot a_t = \frac{1}{\tau_{\text{act}}} \cdot u_t, \]

where \( F \) is the muscle force, \( F_{\text{max}} \) is the maximum isometric force, \( f_{\text{f}} \) is the normalized force-velocity relation and \( f_{\text{l}} \) the normalized force-length relation. \( F_{\text{max}}^{\text{passive}} \) is the maximum passive force and the \( f_{\text{l}}^{\text{passive}} \) the normalized passive force-length relation. \( a_t \) is the muscle activation and \( \tau_{\text{act}}, \tau_{\text{deact}} \) are activation and deactivation time constants set to 20ms and 40ms. \( u_t \) is the motor neuron activation set by the sum of the CPG output and the reflex outputs. (Yakovenko et al., 2004)

The CPG pattern was based on electromyographic profiles derived from a large number of locomotor studies of the simulated muscle groups during slow walking in cats. The swing and stance phases were set to 30% and 70% of the step cycle duration respectively, which other studies have shown is an appropriate relationship for slow walking with cycle times of about 600ms. (Yakovenko et al., 2004) The CPG pattern was fixed and the pattern for the different legs was phase shifted by 50%. An internal clock variable, \( t_i \), specifying the current position of the readout of the CPG-activation pattern, determined what activation the CPG should exert on the actuators. For a schematic of the simulation program, see Appendix II.

The reflexes were modeled using the following activation models derived from the literature (Prochazka, 1999):

Ia model:

\[ f_{\text{la}}(l,v) = K_{la} \cdot \left( 65 \cdot v^{0.5} + 200 \cdot l + 60 \right), \]  

Ib model:

\[ f_{\text{lb}}(s,F) = K_{lb} \cdot F \cdot \frac{(s + 0.15) \cdot (s + 1.5) \cdot (s + 16)}{(s + 0.2) \cdot (s + 2) \cdot (s + 37)}. \]

In (13), \( f_{\text{la}} \) is a time function of the Ia-reflex response for changes in length and velocity. \( l \) is the displacement given in rest length and \( v \) is the velocity given in rest length per seconds. In (14), \( f_{\text{lb}} \) is given in the frequency domain and returns the Ib-reflex response from the tendon organs. The \( K_{la} \) and \( K_{lb} \) are gain constants for the reflexes and were set as to adjust the mean response of each reflex to 15% of the CPG activation throughout the locomotor cycle. Thus, the total addition of the reflexes to the CPG activation is in accordance with studies of walking cats. To comply with parameters observed in other studies, the reflex latency was set to 35ms in all muscles and this feedback was only activated once the CPG activation profile of the muscle was nonzero. (Yakovenko et al., 2004)
The model incorporated a Boolean form of finite state rules. The linguistic interpretation of these rules is stated above (see section 1.4 If-Then rules). The finite state rules used a division of the CPG activation pattern into two phases, Swing and Stance. These phases were set to 30% and 70% of the activation pattern respectively, see Fig. 10 for more detail. The rules were based on the instantaneous forces and lengths of certain muscles, not the delayed lengths and forces used for the reflex responses.

The Stance-to-Swing rule moves the CPG readout position, $t_i$, to the corresponding E4 phase of the step cycle for the specific leg. The rule has three input statements that must be satisfied in order for the rule to fire:

1. The CPG readout time, $t_i$, must be in the Stance interval for the specific leg.
2. The hip-flexor muscle must be over 95% of its maximum length.
3. The absolute force on the ankle-extensor muscle must be less than 5N.

The Swing-to-Stance rule instead moves the CPG readout position to the corresponding start of the E2 phase for that specific leg. This rule also uses three input statements to determine firing:

1. The CPG readout time, $t_i$, must be in the Swing interval for the specific leg.
2. The hip-flexor muscle must be shorter than 25% of its maximum length.
3. The ankle-extensor muscle must be longer than 40% of its maximum length.

In the original model, there was a fixed phase difference between the CPG-activation patterns of the two legs. Thus, when a rule is triggered for the ipsilateral leg this automatically moves both the readout position of the ipsilateral leg as well as that of the contralateral leg. Hence, the firing of the Stance-to-Swing rule for one leg more or less corresponds to the firing of the Swing-to-Stance rule of the other leg.

2.1.1 Modifications to original model

Described above is the original model. This model was modified in three major areas: implementation of separate CPGs for each leg, implementation of new stretch reflex models and the substitution of If-Then rules by an FLC. Here I will describe the two first modifications, the FLC implementation is thoroughly described in section 2.2 Implementation of the FLC below.

The independent CPGs were constructed through duplicating the CPG-activation pattern and giving each leg its own time integrator, $t_i$, determining the position in the CPG-activation pattern. This change does not entail any phase shift between the legs in the model except if If-Then rules shift the activation pattern for the legs separately. Since the original model had a set phase difference between the activation patterns of the two legs, the triggering of If-Then rules switched the activation patterns of both legs simultaneously, hence maintaining the phase difference. Uncoupling the two activation patterns thus introduces a new degree of freedom in the model given that the If-Then rules or the FLC act on each leg independently as in the simulations below.

G. Bosco et al. (1996), as early as 1996, proposed that DSCT-neurons in the spinal cord code the position of the endpoint of the limb in polar coordinates. Later, the angle and length information was shown to be coded separately, (Poppele et al., 2002), and just recently Poppele et al. (2003) showed that this information is modulated by the contralateral limb. Hence, I implemented new outputs from the forward dynamic model, in Working Model, giving MatLab the information of the legs’ endpoints in polar coordinates. As in the original model, stretch reflexes were only active when the muscles were activated by the CPG and the gains of the responses were set to generate approximately 30% of the mean activation of the muscles. This was achieved by running simulations of each model with at least two different pa-
rameter sets and tuning the gains so as to set the mean responses to be approximately the same as the responses for the reflexes using the original reflex models.

2.1.1.1 New model for the Iα-afferent feedback, Global Iα-model

Working Model provides the values of length and velocity of the endpoint of each leg, so I thought it would be interesting to implement an entirely new and simplified means of Iα-afferent feedback to each muscle based on the length of the whole leg. The old Iα-reflex model acts on each muscle separately and uses the displacement and the velocities of each muscle separately expressed in rest lengths (RL and RL/s). The rest length of a muscle is the length of the muscle when not contracting or stretched. Usually this length is near the mean of the maximum and minimum length of the muscle. The displacement is defined as the length of the muscle additional to its minimum length, expressed in RL.

The new reflex model essentially views the whole leg as having two fictive muscles, one extensor and one flexor, and lets the Iα-reflex model work on the displacement and velocity of the whole leg instead of each muscle separately. The response of the reflex model is then tuned by a separate gain for each muscle. The RL of the leg was determined as the mean of the maximum and minimum distance between the hip and the leg endpoint (the toe). The displacements and velocities were defined as follows. Flexor muscles are the shortest when the leg is flexed. They lengthen as the leg extends and the distance between hip and endpoint increases. Therefore, flexor muscle displacement was defined as the difference between the current and the minimum distance from hip to endpoint and the velocity was defined as positive if this difference increased. Extensors are at minimum length when the leg is extended and they lengthen as the leg is flexed. Thus, extensor muscle displacement was expressed as the difference between the maximum and current distance between hip and endpoint and velocity was defined as positive if this difference increased.

The lengths and velocities were used in (13) for the fictive flexor and fictive extensor separately to generate the fictive extensor and flexor responses and respectively. The gain constant vector was tuned by trial and error to scale the Iα-reflex component of response of each muscle separately, so as to result in about 15% of the mean activation of each muscle as was the case in the original Iα-model.

2.1.1.2 New model for the positive force feedback I, GRF Iβ-model

In addition to the simplified global Iα feedback just described I also used a single force variable, the component of ground reaction force (GRF) directed from toe to hip, to exert a positive load feedback to all extensor muscles individually. GRF was limited to the extensor muscles because only they are active during Stance. Since the flexors are usually only activated when the leg is off ground, they have negligible effect on the GRF so I chose not to provide flexors with GRF-related feedback.

As just stated I chose to use the component of the GRF projected onto a vector directed from the toe to the hip, since the GRF during Stance usually points in that direction (see Fig. 17 b,c). This input was filtered through (14) and the gain, \( K_{Iβ} \), set for each extensor muscle to correspond to about 15% of the mean activation as in the original model. For the flexor muscles the activation from the Iβ-reflex was set to zero.

2.1.1.3 New models for the positive force feedback II, Summed Iβ-model

This model used a similar holistic view of the leg as the models above. However, as the title implies, instead of using the global GRF in this model I used the sum of the Iβ-reflex responses of all the muscles as a single input variable for positive load feedback. The original Iβ-transduction model, (14), was used to convert the forces in the muscles to the individual Iβ responses in each muscle. These were then summed for extensors and flexors separately. As
in the GRF Ib-model the gain vector, $K_{ib}$, scaled activation to generate about 15% of mean activation, as in the original model.

### 2.2 Implementation of the FLC

The Fuzzy Logic controller, FLC, was designed to determine the switching between the stance and swing phases for the individual legs separately. Thus, a FLC was implemented that took the inputs for one leg at a time and determined the switching of that leg, for the FLC see Fig. 11. There were two switching rules, one from Stance to Swing and vice versa. In order to simplify the fine-tuning of the rules, they were implemented as two separate outputs. Since this is a Boolean condition, the outputs of the FLC were compared to a threshold and reaching that threshold initiated the switch by moving the readout position of the CPG-activation pattern, $t_i$, for that leg. The specific threshold was irrelevant since the outputs of the FLC could be tuned to conform to any threshold. The thresholds were set arbitrarily to 1.15.

The Fuzzy Logic Controller, FLC, was implemented using MatLab’s Fuzzy Logic toolbox. The choice of using MatLab and its toolboxes was based on the ease with which the interface to the old simulation code could be constructed given that the time constraints of the project did not permit the construction of a comparable user interface. The toolbox incorporates two important functions, `fuzzy()` and `evalfis()`. The first is the FLC viewer and editor. This user interface was used to design the structure of the FLC, construct the fuzzy rules, tune membership functions and tune the weights of each rule. The `evalfis` function is used to calculate the output of the FLC given a certain input vector. For every time increment in the simulation, Working Model 2D calls a function in MatLab that returns the forces of the actuators in the two legs, see Appendix II. For every call, MatLab determined if a switching should occur for each leg. Hence, the `evalfis` function is called twice, one for each leg, for every Working Model 2D call.

The MatLab fuzzy toolbox has the option of implementing either the Sugeno or the Mamdani inference system. I chose to use the Mamdani min-max inference system for the FLC since it is the more intuitive of the two. As stated above, this type of inference system is more computationally demanding than the Sugeno-inference, but since there was no demand for the simulation to run in real-time this was not viewed as a critical criterion for the FLC.

For the choice of defuzzification method, there are several criteria to take into account. Some of them are stated in section 1.5.5 The control system. The MatLab fuzzy toolbox gives the possibility to use several different methods of defuzzification and among them is the CoA, CoS- and MoM-method. To facilitate the fine-tuning of the FLC, the MoM was discarded since it has both a discontinuous input-output relationship and using scaled inference the plausible criterion is the same for the two methods. The CoA-method was discarded since it requires that the output membership functions should not overlap completely for all rules given that they should all contribute to the output at all times. Therefore, I used the CoS-method of defuzzification in the experiment, which also is less computationally demanding than the other methods.

As stated above the FLC has two outputs. With the use of the CoS-defuzzification method, these outputs are able to vary continuously over the output space. The output space was set...
for both outputs to [0, 2]. In order to make fine-tuning of the controller easy, only two output membership functions were present in the output space. These output membership functions were implemented as symmetrical triangular functions with a width of 0.2 and were symmetrically placed around 1. For more specifics details, see Appendix III.

When the inputs for the FLC were to be chosen, I looked at several different options. Dependent on the choice of inputs, different If-Then rules might be implemented. The rules stated in section 1.4 If-Then rules do not take into account the fact that the spinal cord can integrate information from several proprioceptors, as is done by the DSCT-neurons (Bosco and Poppele, 2001). The DSCT-neurons code the leg endpoint-position in polar coordinates for the length and the angle independently (Poppele et al., 2002). Given that the FLC should enhance the stability range, for the parameters $F_{\text{max}}$ of the model and given that certain configurations of $F_{\text{max}}$’s could give gaits where the hip extensor muscle length does not correlate with the possibility for the leg to give postural support, I chose to develop rules based on these integrated inputs. The FLC was given three inputs that reflected the integrated information: the angle of the hip to foot vector, the length of the leg and the load on all the extensor muscle of the leg. For the switch from Swing to Stance, the total load on the extensor muscles has a peak when the retractors fire at the end of the swing phase. Since this force peak is significant, I chose to use the ankle extensor muscle Ib-response instead as load variable for the Swing–to-Stance transition.

The firing of DSCT-neurons is also modulated by input from the contralateral leg and the firing of DSCT-neurons is modulated for both in-phase and out-of-phase movements of the two legs, (Poppele et al., 2003). In our model the read-out speed of the CPG combined with the $F_{\text{max}}$ are set as to enable an out of phase gait by the model. Therefore, I chose to implement this linkage by giving the FLC the contralateral leg’s FLC’s StSw-output. This output, as shown below, is dependent on the load and the angle of the contralateral leg. Furthermore, I know that the rules are state-dependent, see section 1.4 If-Then rules, thus to incorporate this in the model the FLC needed a measurement of phase. I chose to use the variable $t_i$ for the measurement of phase, as this variable represents the readout position in the CPG-action pattern for each leg.

Even though only two Boolean If-Then rules were to be converted into the fuzzy logic domain, the use of fuzzy logic vastly increases the number of possible tuning parameters and with it the complexity of the fine-tuning process. For every fuzzy rule, the possible tuning parameters are all of the input and output membership function parameters used by the rule, and the weight of the rule. Therefore, it is important to restrict the number of membership functions to enable an easily tuned FLC. However, the number of membership functions should not be too low, as this could limit controllability. I found that two membership functions on all inputs, for every Boolean If-Then rule, were enough to have adequate controllability of the output. The final membership functions are presented in Appendix III. Since I chose to limit the number of membership functions, I also avoided the predicament of an exponential growth of rules with respect to membership functions described in (Novakovic, 1999).

2.2.1 Rule base

During the development of the FLC, several different rule bases were implemented. All rule bases were developed using the Boolean rules in section 1.4 If-Then rules as templates. However, they were modified for the use of the global inputs described above and a degree of inter-leg dependence was introduced with the use of the contralateral leg’s FLC output as one input. This was done in acknowledgement of the well-known interlimb coupling demonstrated in real cats.

Given the nature of the inference system, each Boolean rule has to be represented by more than one fuzzy rule in order for the FLC not to act as a Boolean trigger. Therefore, the two
crisp rules were implemented as at least one rule ‘voting’ for a switch of phase, and usually
one rule for each input voting for staying in the current phase. This implementation thus re-
resulted in a weighing of the rules voting to switch and the rules not voting to switch.

The importance of each rule is tuned by a scaling factor. Since the output membership
functions are of equal area and the number of rules voting not to switch is greater than the
number of rules voting to switch, the scale factor for the rules voting not to switch were set
much lower. Naturally, the scale factor and the input membership functions for the rules are
highly interdependent and contribute as a whole to the FLC output. This has to be kept in
mind when viewing the rule bases and membership functions, but generally speaking, the
higher the scale factor the more importance this rule has in determining the output. The final
rule base is presented in Appendix IV.

2.2.2 The fine-tuning of the FLC

The fine-tuning of the FLC was performed by having a set of simulation parameters that re-
resulted in both stable and unstable gaits without any rules. By running the simulations on this
set of parameters with the FLC active, instances where the FLC failed to act appropriately
were gathered. By assessing the inputs and the output of the FLC, I changed one or two pa-
rameters at a time. As the number of membership functions and rules were kept to a mini-
mum, the tuning process was held reasonably efficient.

The goal of the fine-tuning procedure was to stabilize a subset of test parameter set. The
causes of the limitation in the fine-tuning process were:

1. Controlling the transition in readout position may not stabilize all sets of $F_{\text{max}}$.
2. Any alternation of the CPG-activation pattern readout both shifts and resized the
   stable region in six-dimensional parameter space.
3. The time constraint of the project did not permit an extended search.

2.3 Parametric testing routine

In order to compare variation in stability between the different controller designs, a paramet-
ric testing routine was set up. This routine, implemented for the original model by Yakovenko
et al. (2004), expressed the forces in the musculotendon actuators as a parametric space, $F_{\text{max}}$
in (11). Since there were six actuators in the model, the parameter testing routine sought vol-
umes in this six-dimensional space associated with stable locomotion.

The measurement of stability was defined as the time that the model walked without falling
during the simulation. In order for the parameter to be classed as belonging to the stable vol-
ume, the model had to walk for 12s, equivalent to approximately 20 steps. The random search
routine, implemented by Yakovenko et al. (2004), was used to find stable points in the para-
metric space by starting at a random point and following the gradient of the simulation time.
This random search routine obtained, by following the gradient, a number of unstable points
and stable points on the rim of the stable volumes, unless by chance the random starting point
was inside a stable volume.

The random search routine ran independently on several personal computers. Thereafter,
the data from all the computers were gathered and further analysed. Since the random search
routine used random starting points, the risk arose that the same point would be evaluated by
different computers. In order to remove any bias that duplicate values may give, all duplicates
were removed before analysis.

2.4 Evaluation principle for Results

The stability of the different control mechanisms was evaluated by comparing required a comparison of the range of force parameters producing stable locomotion. The random search routine returned a number of simulations, some stable but most not. Each simulation was associated with an individual set of $F_{\text{max}}$ for its muscles. The sets of $F_{\text{max}}$ that produced a 'stable' gait formed a cloud in the six dimensional parameter space. The size of this cloud provided a measure of overall stability of the given control mechanisms.

In principle, quantifying cloud size could be done in several different ways. The method that first comes to mind would be to measure the six dimensional volume defined by the extent of the stable data points, but since there is no evidence that this volume would necessarily be convex, this approach becomes too complex. The following formula was used by Yakovenko et al. (2004) to quantify the volume:

$$ I = \sum_{i} \frac{|r_i|^2}{n}, \quad (13) $$

For a convex volume, this can be a good quantification. However, the quantification fails when more than one volume is stable or when the volume is nonconvex, see Fig. 12. The formula (13) overemphasizes outlying data points.

As any measurement based on the radius from a mean to the scattered data points will attribute outliers too much importance, I developed a new measurement strategy. This computed the percentage of stable points in a random subset of validation points in the six dimensional space. The validation points were classified as stable according to a K-nearest-neighbour algorithm. The algorithm gathered a fixed number (10,000) of classified validation points. Thus, the measurement was independent of the shape of the stable volume.

A way of visualizing these data is to perform a principal component analysis (PCA) on the stable points. This analysis extracts the orthogonal directions in which the data have the highest variances. Mapping the data onto the principal component coordinate system can give us a rough estimate of the extent of the stable volume.

Fig. 12 a. Shows a two-dimensional convex set. b, c. illustrates the error involved with measurements based on distances from a mean. b and c have the same area but a distance based measurement will favour c.
3 Results
Since the results concerning the Fuzzy Logic Controller, FLC, are unrelated to the new proposed global reflex models, I have chosen to separate these two items. Furthermore, for comparison I will include the I-measurement of the stable cloud as used by Yakovenko et al. (2004) for all simulations. All data referred to below, unless otherwise stated, were gathered using the parametric search routine described in section 2.3 Parametric testing routine.

3.1 Effect of the FLC
In order to compare the effect of introducing a FLC, we ran two sets of simulations. The first set incorporated central activation and stretch reflexes but no FLC, the CPGSR set. This was the model developed and studied by Yakovenko et al. (2004). The second model added FLC to CPGSR to trigger phase transitions. Since the results are expressed in the six dimensional space of the maximum force of the musculotendon actuators, $F_{\text{max}}$, viewing these data presents a problem. Using PCA to extract the two most important PC’s and plotting the data points in this new coordinate system, I was able to portray the stable volume in an intuitive way. However, reducing the number of dimensions to two neglects some of the variance in the data, therefore the stability criteria were based on all available dimensions.

As seen in Fig. 13 a-b, the FLC affected all but four of the stable simulations. These four instances of the stable gait ranged from barely physiological to very abnormal gaits. The simulation setup with no rules, CPGSR, had numerous stable regions. In other simulations where the same points were assessed for both the control mechanisms in FLC and CPGSR set, the FLC triggered for the majority of these points. Even though the FLC at these points does not contribute to the stability of the model, one way of viewing the CPG is as an oscillator with a low frequency that by proprioception adjusts its speed to suit the locomotor situation at hand. Through the change in readout position the CPG-activation pattern is recalibrated to more properly suit the situation. This can be viewed in Fig. 14 showing a perturbed gait.
In the final FLC rule base (see Appendix IV), the importance of the interlimb coupling can clearly be seen in the Swing-to-Stance rules. Among the rules with the highest weights is the rule voting for change of phase dependent on whether the other leg’s Stance-to-Swing rule is nearing its threshold for firing. This rule thus tends to promote a phase difference between the legs by nearly always making sure that the FLC triggers the Swing-to-Stance rule before the Stance-to-Swing rule for the other leg. When the CPG-activation pattern changes phase from Stance to Swing the postural support role of that leg is lost since the leg will start to retract. I have noticed that a transition from Stance to Swing before the contralateral leg is in E1 or end of Swing, usually leads to a loss of postural support leading to a crouch or instability. Changing phase of the contralateral leg to Stance before the loss of postural support from the ipsilateral leg thus prepares the leg for weight-bearing by stiffening the muscles. Therefore, I believe that the transition from swing to stance phase of the contralateral leg before the ipsilateral leg’s transition from Stance to Swing helps to stabilize the model.

An early Swing-to-Stance transition was not destabilizing, provided that the foot was further forward than the hip. In such cases, the weight-bearing function was transferred to this leg, reducing the load on the other leg, which therefore changed to the swing phase earlier.

Controlling the Stance to Swing transition was implemented using two rules for initiating the Swing. According to most of the literature, the Stance-to-Swing transition is controlled by two factors, the load on ankle/leg and the angle of the hip joint. I constructed two fuzzy rules to mediate the transition. The first rule is merely a fuzzification of the classical rule (see Appendix IV first Stance-to-Swing rule). This rule uses the summed extensor muscle forces of the leg and the angle between the body and a vector from hip to foot. I found that using the overall load on the leg gave a better approximation of when the leg no longer contributed to postural support than only viewing the load on the ankle extensors. This might be because many of the simulations involved a crouched gait, in which ankle extensor force is poorly correlated with ground reaction force. The second rule used only the load on the whole leg (see Appendix IV second Stance-to-Swing rule). This rule combined with the first one results in a switch of phase either because of loss of load on the leg or because the foot is so far back and the load is too low for the leg to be able to contribute to the postural support.

The FLC was associated with a wider spread of stable speeds as compared to only the CPG and stretch reflexes model, see Fig. 15 c. In the latter case the CPG had a set period of 0.6s, which forced the step cycle to have a constant period irrespectively of the forces produced by the muscles. This limited the models stability range to a narrow band in the speed domain, since either too high or too low muscle forces would introduce mechanical work not suited for this cycle period.
FLC broadened the stability range in the speed domain by allowing variable periods. By skipping parts of the activation pattern, FLC control resulted in shorter periods which were better matched to the mechanical requirements. The range of stride lengths in the two models was similar because stride length is constrained by the structure of the model and the fact that the model does not change gait to trot or gallop. Hence, the wider diversity in velocity is introduced as a spread in velocity over a specific stride length, see Fig. 15 a,b.

### 3.2 Effect of the global Ia- and Ib-reflexes

The effects of the new reflex models were evaluated through running two sets of simulations global reflexes. The first set, henceforth referred to as WholeGRF, used the Global Ia-reflex and the GRF Ib-reflex described above. The second set of simulations, WholeIaIb, used the same Ia-reflex model as the first combined with the Summed Ib-reflex. These two sets of simulations were compared to the control model, CPGSR, in which muscles were individually controlled by their own stretch reflexes as implemented by Yakovenko et al. (2004). As can be seen in Fig. 16, the \( F_{\text{max}} \) of the stable simulations for all of the simulation sets overlap considerably. The stable points in the set WholeGRF largely overlap the CPGSR set which indicate that the lack of Ib-reflex response for the flexors does not significantly change the extent of the stable region.

Evaluating the three simulations sets on the plane of the first and second PC of the CPGSR simulation set gave a number of points where all three sets were stable. However, a comparison of joint angle profiles in the step cycles of these simulations showed some discrepancies between the models. As can be seen in Fig. 17 a, the joint angles for most of the step cycle were not significantly different between all the reflex control mechanisms. However, the WholeGRF set differed from the others at the end of the stance phase. From mid-Stance to end-Stance the ankle and knee joints did not extend as much and thus the WholeGRF model had a more crouched gait. This was due to a lower activation of the muscles by the reflexes at the end of the Stance. The main cause of this is that the GRF Ib-reflex was not given the same mean activation as the Ib-reflex in CPGSR. How-
ever, to some extent the Global Ia-reflex compensated for this loss in activation and brought the mean activation of both the Ia- and Ib-reflex activation to approximately 30% of the mean muscle activation, but the mean activation for the CPGSR was, in the first and second PC-plane, nearly 40%. As can be seen in Fig. 17 b-c, the timing of the peak activation for all the muscles differed significantly between the reflex models. With the two global reflex models the subtle difference in timing of the response to load and stretch each of muscles was eliminated, as only one overall compensatory control variable was fed back.

Even though the activation levels of the WholeGRF random search set were significantly lower than for the CPGSR, the mean $F_{\text{max}}$ values of all the actuators were quite similar and, furthermore, the lowest extensor activation levels were similar (see Fig. 16). Yakovenko et al. (2004) compared the CPGSR model to the CPG model without reflexes. In the latter, the extensor $F_{\text{max}}$ values were significantly larger since more central activation was needed in the absence of some activation coming from feedback. Therefore, I must conclude that even if the proportion of the muscle activation fed back by the WholeGRF reflexes for the extensors is much lower it contributes significantly to gain stability when central activation is low.
The spread of mean velocities for the different reflex models was comparable (Fig. 18 a). However, the CPGSR set had a wider peak, and thus its stability was not as restricted over the range of velocities. The global reflex models both show two peaks in their distributions, the first at 0.4m/s and the other at 0.7m/s. This gives their velocity distribution a more selective nature. However, they do span approximately the same region. Noteworthy is that the range of stability of the WholeGRF is shifted towards lower speeds, as is that of the CPG model. This is presumably related to the lower mean activation levels of the reflexes (0% in CPG model, 20% in WholeGRF, 38% for CPGSR and 35% for WholeIaIb).

The global reflex models also had narrower peaks in their step length distributions (Fig. 18 b). However, they also had a wider base of stability than both the CPG set and the CPGSR models. Since the mechanical model limits the maximum step length it is not surprising that all simulation sets have the same maximum step length.

In an attempt to quantify the spread in walking patterns between the different reflex models, I analysed the mean and standard deviation of the joint angles in the step cycle. Keep in mind that the data points represent different $F_{\text{max}}$ values and thus analysing the whole data sets will not produce exactly the same forces in the actuators. However, the standard deviation around the means of joint angles in the step cycle should give an idea measure of the spread in walking patterns. Interestingly we found no significant difference in the standard deviation of step-cycle joint angles between the reflex models.

### 3.3 Stability comparison

The stability of the different models were compared using three different techniques. To allow a direct comparison with the results of Yakovenko et al.’s study (2004), I-values are presented in Table 2. Yakovenko et al.’s (2004) simulations of the CPG alone scored an I-value of $3.0 \times 10^4 \text{N}^2$, and therefore according to this measurement all the new reflex models scored higher. However, according to the I-value stability was no greater under FLC than CPGSR set.
Fig. 19 a-f, shows sets of simulations run on the plane of the first and the second PC of the CPGSR simulation set. From these simulations it is obvious that the form of the stable points was neither a convex cloud nor even a single continuous volume of stability. This test also indicates that the stability of the FLC greatly exceeded that of central activation with stretch reflexes. The stability of the different reflex models was fairly comparable. Fig. 19 e-f, shows the finite state (crisp If-Then rules) model implemented by Yakovenko et al. (2004). The coupled simulation was identical to that used by Yakovenko et al. (2004), whereas in the uncoupled simulation the If-Then rules controlled the legs independently. As with the other results, Fig. 19 e-f clearly shows that at least in this plane the finite state control were less stable then the FLC control and that the stable volumes of the If-Then rules were not convex.

Fig. 20 a-d is an attempt to visualize the stable volumes in the three most important PCs of the different simulation sets, which together accounted for 75-80% of the variance in the sets of data independently. Therefore, all the points in this space would be more scattered if the three remaining PCs were added. Using a K-nearest-neighbour classification, with K=1 and with a distance limit of 40N, the PC spaces of the simulation sets were evaluated in the planes of the first and second PCs. This approach to the analysis clearly revealed that the reflex models have scattered stable volumes surrounded by unstable regions. The greatest improvement for the FLC set was that the stable volumes were more closely packed which in turn gave them a more convex appearance. Unfortunately, because of the time constraints a random search analysis could not be performed on the coupled and uncoupled finite state (If-Then) models.

The K-nearest-neighbour and the I-value stability measurements (Table 2) gave very different views of the stability gained by the FLC. Using K=1 and having no distance restraint, the K-nearest-neighbour measurement gave a similar relationship between the reflex models as did the I-value measurement. However, when it came to FLC the K-nearest-neighbour measurement showed a significantly enlarged region of stability. Therefore, I must conclude that the FLC greatly improves stability and that global reflexes models have a comparable stability to traditional stretch reflex models.

<table>
<thead>
<tr>
<th>Simulation set:</th>
<th>CPGSR</th>
<th>WholeGRF</th>
<th>WholeIaIb</th>
<th>FLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-value (N²)</td>
<td>6.2x10⁴</td>
<td>4.9x10⁴</td>
<td>5.1x10⁴</td>
<td>6x10⁴</td>
</tr>
<tr>
<td>K-nearest neighbour</td>
<td>2.3%</td>
<td>3.8%</td>
<td>4.6%</td>
<td>41.8%</td>
</tr>
</tbody>
</table>

Table 2. The scalar stability measurements.
Fig. 19 a-f show the stable area of the different simulations sets on the first and second principal components of the CPGSR set. The numbers by the contour lines indicate the time in seconds from onset until the model fell. All simulations were stopped after 13s and the model was classed as stable if it walked for at least 12s.
Fig. 20 a-d. This shows the classification of stability based on nearest neighbour, black stable and grey unstable. Note the scattered groups of stable points surrounded by unstable points. Each plot is made in the first, second and third principal components of the stable points in the search data, which accounts for 74%, 80%, 83% and 75% of the variance respectively.
4 Discussion

Yakovenko et al. (2004) showed that the use of finite state rules to fine-tune CPG-activation patterns in a locomotor model improved its stability. It has been argued that fuzzy inference systems are not only biologically plausible, but they might also provide additional stability in locomotor control (Prochazka, 1996a; Bosco and Poppele, 2001). The present study has shown that when a FLC was used to determine the transition between stance and swing phases, stability was indeed improved. FLCs use a form of weighing of multidimensional sensory inputs, much as neural nets do. Because they do not operate on a strict Boolean basis as finite state controllers do, but rather on scaled responses related to sensory domains, it has been argued that FLC represents a linguistic interpretation of the neural machinery that controls phase transitions. Admittedly, since the design of a FLC involves numerous experience-based qualitative and quantitative decisions, for example the choice of inputs and their membership functions, the fuzzy representation may not correspond in all respects to neural mechanisms. Nevertheless, this study has shown that FLC is clearly capable of greatly improving stability and therefore similar control rules may be of importance in real cats or may also provide interesting solutions for locomotor control of legged robots.

Global inputs were used for the FLC, such as limb orientation and a summed extensor muscle load. It would be interesting to compare the stability of phase controllers based on ‘local’ variables, such as hip extensor length, as opposed to the ‘global’ variables used in this study. However, the time constraints on this project limited the study to the ‘global’ inputs. Furthermore, I would suggest for further FLC analysis that an automatic fine-tuning algorithm should be developed. The manual fine-tuning process limits the complexity of the controller and therefore the possible controllability of the FLCs.

Since our FLC control generated a Boolean transition in the CPG-activation patterns, one might ask whether a continuous adaptation of the readout speed would be plausible. I implemented a form of FLC that did continuously alter the CPG-time integrator in order to stabilize the step. This FLC was far more complex than the FLC implemented in the report above, and this led to it being abandoned. The complexity involved, made manual fine-tuning impossible. Machine learning techniques, for example genetic algorithms, might be able automatically to tune the FLC’s parameters to fit certain locomotor movements. However, since this entails that the FLC, to some degree, tunes the readout rate according to the mechanics of the model, the need for mechanical invariants over all plausible stable gaits arise. Without such invariants, the controller would not be general, and thus it would make erroneous control decisions, limiting the stable points to a predefined walking pattern.

In the parametric testing routine, an algorithm was incorporated for minimizing the number of duplicate points simulated on different computers. However, some problems were encountered in the random distribution of starting points in the random search routine for the reflex model simulation sets: CPGSR, WholeGRF and WholeIaIb. This led to a more restricted search of the parameter space and that a number of duplicate data points were found and removed before the data analysis. Because of the defect in random distribution of starting points, fewer stable data points were left for these simulation sets after the duplicate data points had been removed. Hence, there may have been stable volumes that were not found with the random parameter assignment. However, this should not affect the results significantly, since the stable volumes of the reflex models were small and surrounded by unstable areas, see section 3.3 Stability comparison.

The stability measurements showed that global reflexes had comparable ranges of stability to the ‘local’ reflexes. From a neurophysiological point of view, there is solid evidence that the information needed for global Ia-reflexes is present in the mammalian spinal cord.
The existence of global Ib-input is more speculative, though data on convergence of Ib-inputs onto spinal interneurons do indicate that the Summed Ib-reflex model is biologically plausible. If global reflexes exist in animals, it is likely that they conferred an evolutionary advantage. This study did not find any significant improvements in walking patterns or stability among the various reflex models, including global reflex models, at least in our limited analysis. Further analysis may yet reveal advantages or disadvantages of global reflexes with respect to both input parameters and the tuning of gains. From a robotics perspective it is clear that control based on just one pair of sensors located at the limb endpoint is far simpler (and cheaper) than individual control of several actuators. However, considering that the inputs are present in the spinal cord and that our simulations show that global reflex models can be just as stable as distributed ones, it is now up to the physiologists to prove or disprove the existence of global reflexes.
References


Appendix I
Preparations used in cat and other animal experiments

There is a variety of common preparations used in the study of locomotion. The spinal preparation is a transection of the spinal cord made in such a way that the nerve centres controlling the hind limbs are intact but disconnected them from the cortex and the descending pathways from the forelimb areas of the spinal cord. (Kandel et al., 2000) This preparation allows the investigation of lumbosacral neural circuits controlling rhythmical locomotor patterns of the hindlimbs in the absence of descending influences.

The decerebrate preparation is produced by a transection of the neuraxis at the level of the midbrain. This removes the influence of the cerebral cortex, thus allowing the study of the role of the cerebellum and the lower part of the brainstem in association with spinal pattern generators. (Kandel et al., 2000)

Deafferentation is a preparation where the sensory axons are severed. Thus, all the somatosensory information to the spinal cord is lost. Several important findings have resulted from this preparation:

1) Basic ability to produce voluntary force and movement is preserved, but movements are generally uncoordinated and inaccurate.
2) Coordination of the primate hand segments is particularly impaired. Spatial orientation, fractionated movements and anticipatory preshaping all suffer reduced accuracy.
3) Gait is possible but tends to be irregular and uncoordinated in vertebrates as well as in invertebrates. In humans the gait is severely impaired and needs conscious attention. With neck proprioception lost, gait becomes virtually impossible for humans. (Prochazka, 1996b)
Appendix II
Schematics of the simulation program

Forward dynamic model
Simulates the mechanics of the model.

CPG module
Gets the CPG-activation for each leg based on their separate CPG-time, ti, and updates the CPG-time ti for each leg.

Muscle model
Calculates the passive and active forces for the muscle model.

Ia model
Calculates Ia-reflex response for the muscles.

Ib model
Calculates Ia-reflex response for the muscles.

Delay

FLC
Evaluates the state of each simulated leg in the model and moves readout, ti, when firing.

Working Model 2D

MathLab

\[ x, \frac{dx}{dt}, \frac{dF}{dt} \]

\[ F, ti, K_{Ia}, K_{Ib}, K_{CPG} \]

\[ x, \frac{dx}{dt}, E \]
Appendix III
Membership function

The name in bold is the domain of discourse for the input or output to the FLC. Every row is a membership function in that domain.

**Input membership functions**
The input membership functions are trapezoids. The inf-sign means outside the input range of the domain in the FLC. The trapezoids are defined as:

\[
f(x) = \begin{cases} 
0, & x \leq a \\
\frac{x-a}{b-a}, & a < x < b \\
1, & b \leq x \leq c \\
\frac{b-x}{c-b}, & c < x < d \\
0, & x \geq d 
\end{cases}
\]

<table>
<thead>
<tr>
<th>Angle2Foot</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>SwSt_Sw</td>
<td>-inf</td>
<td>-inf</td>
<td>55</td>
<td>87</td>
</tr>
<tr>
<td>SwSt_NoSw</td>
<td>50.6</td>
<td>89</td>
<td>Inf</td>
<td>inf</td>
</tr>
<tr>
<td>StSw_Sw</td>
<td>100</td>
<td>165</td>
<td>Inf</td>
<td>inf</td>
</tr>
<tr>
<td>StSw_NoSw</td>
<td>-inf</td>
<td>-inf</td>
<td>106</td>
<td>146</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LoadAnkle</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>SwSt_Low</td>
<td>0</td>
<td>0</td>
<td>0.16</td>
<td>0.45</td>
</tr>
<tr>
<td>SwSt_High</td>
<td>0.3</td>
<td>0.45</td>
<td>Inf</td>
<td>inf</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LoadOnLeg</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>StSw_Low</td>
<td>0</td>
<td>0</td>
<td>0.07</td>
<td>0.69</td>
</tr>
<tr>
<td>StSw_High</td>
<td>0.19</td>
<td>0.61</td>
<td>Inf</td>
<td>Inf</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OtherLeg</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
<td>1.03</td>
</tr>
<tr>
<td>High</td>
<td>0.92</td>
<td>1.1</td>
<td>Inf</td>
<td>inf</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ti</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swing</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Stance</td>
<td>30</td>
<td>30</td>
<td>85</td>
<td>85</td>
</tr>
</tbody>
</table>

**Output membership functions**
The output membership functions are triangular, with the following set of parameters.

<table>
<thead>
<tr>
<th>StSw</th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSwitch</td>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>Switch</td>
<td>1.4</td>
<td>1.5</td>
<td>1.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SwSt</th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSwitch</td>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>Switch</td>
<td>1.4</td>
<td>1.5</td>
<td>1.6</td>
</tr>
</tbody>
</table>
Appendix IV
The rule base

The rule base is the linguistic interpretation of the fuzzy inference system. Below I state the nine final fuzzy rules that make up the rule base of the FLC. To the right of the rules is an example of the inference system in action. The vertical lines symbolize the crisp input values. The fill of the membership functions in each row represent their truth-value. To the far right is the inferred output of the rule and these outputs are the aggregated at the bottom right where the defuzzified crisp output is shown by a vertical line. For a better overview, the FLC rules have been divided so that they represent only one transition rule.

**Stance-to-Swing rules:**

<table>
<thead>
<tr>
<th>Rules</th>
<th>weight</th>
<th>Angle2Foot = 79.1</th>
<th>ti = 0</th>
<th>LoadOnLeg = 0.346</th>
<th>OtherLeg = 0.896</th>
<th>StSw = 1.04</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Swing-to-Stance rules:**

<table>
<thead>
<tr>
<th>Rules</th>
<th>weight</th>
<th>Angle2Foot = 127</th>
<th>ti = 50</th>
<th>LoadOnLeg = 0.504</th>
<th>StSw = 0.886</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>