A Bio-Inspired Robot Controller

Fabrizio Patuzzo

Master Project

Assistant: L.Righetti

Professor: A.Ijspeert

Biologically Inspired Robotics Group (BIRG)

October 2, 2006

To my nephew Xavier and my niece Claire

Acknowledgements

I would like to thank especially my family: my mother, my father, my brother, his wife, my nephew and my niece for their affection. A special thank goes to my friends at the EPFL: Andrea Krpo, Thomas Mueller, Gabriele Prosperi, Nicolas Genko, Maria Trias, Giulia Chemolli, and Arianna Bosio. I am also deeply grateful to Giorgio Brambilla, who has unfortunately left the EPFL.
I would like to thank my assistant, Ludovic Righetti, and my supervisor, Auke Ijspeert, for their impressive patience and regard during the project. I am grateful to all the members of BIRG for their help, and in particular to Alessandro Crespi for introducing me to the ABBA group that was a pleasure to listen to while running simulations, and for helping me with all sorts of technical problems.

Contents

List of Figures

1 Introduction

1.1 Motivation

Developing controllers for biped robots is one of the great challenges of 21st century robotics. Beside being fun, this research is expected to find applications ranging from service robotics and entertainment robotics to the exploration of terrains inaccessible to humans (atomic nuclear plants, distant planets, etc.). Compared to wheeled robots, legged ones do not require a continuous support path and, hence, can move on a wider range of terrains. Beside being useful to build robots, research on the principles of biped locomotion is important to other fields, including prosthetics (to develop prostheses for handicapped people), sports science (to improve athlete's performance), neuroscience (to understand the human nervous system), and psychology (to understand people's behavior).

1.2 Objective

The objective of this project is to improve a bio-inspired robot controller, by introducing sensory feedback. Ultimately, our enhanced controller should allow the robot to make turns, speed up, slow down, and adapt its gait to various terrain conditions. The project is carried out in simulation, using Webots as a simulation environment. It is written in C++, and uses the Open Dynamics Engine (ODE) to implement the foot sensors. The original controller was written by Ludovic Righetti at the Biologically Inspired Robotics Group (BIRG), Ecole Polytechnique Federale de Lausanne (EPFL).

1.3 On Biped Locomotion

All bodies in our universe seem to follow, to a great extent, Newton's three laws of motion. Animals and robots make no exception, and, hence, to understand how they can move in the world it might be useful to have a quick look at these laws:

1. An object at rest or traveling in uniform motion will remain at rest or traveling in uniform motion unless acted upon by a net force.
2. The rate of change of momentum of a body is equal to the resultant force acting on the body and is in the same direction.
3. All forces occur in pairs, and these two forces are equal in magnitude and opposite in direction.

[I.Newton, *Philosophiae Naturalis Principia Mathematica*, 1687.]

Assuming the body (whether animal or a robot) is initially at rest, the first law implies that to start moving, a force must accelerate it. Since our body moves autonomously (the only external forces are gravity and inertia), and assuming that gravity is unalterable, the only way to start moving is to exert a force on the ground. By Newton's third law, the ground will exert an equal and opposite force on the body, making locomotion possible. Of course, we are assuming that air does not provide sufficient friction, and that the body cannot propel itself forward using a thruster. Summing up, to start moving an animal or robot must exert a force on the ground.

Due to friction in the environment, exerting a force on the ground once is not sufficient to keep moving: the animal would quickly slow down (or fall) and stop. How do animals solve this problem? Apparently, most animals, whether they move on the ground, in water or in air, have come up with the same solution: exerting rhythmically forces on the environment, over and over in the same way. This is the way elephants amble, horses trot or gallop, birds fly, and fishes swim: they rhythmically move their limbs, exerting forces on the environment.

In this project, we are interested in *biped* locomotion. Hence, we will assume that our animal or robot exerts forces to the environment using its legs, reserving its upper limbs for other useful tasks because this is the way biped animals walk. Using only the lower limbs to move, however, makes the problem of maintaining balance more difficult: to be able to walk on and on, a biped animal should prevent falling on the ground. Human beings are very good at this, and learn to maintain balance when they are about 1 year old. Having a biped robot do the same, however, is much more difficult, and is actually the greatest obstacle toward developing biped robots. As Wieber [10] nicely put it:

> Observing that falling induces a significant risk to definitely disrupt any possibility to achieve any objective at all (in case of a major breakage, for example), and that an objective that cannot be achieved at a given moment can usually be postponed without particular contraindications, we can conclude that in the general case: The major issue for walking systems is to avoid to fall, and more precise objectives can be taken care of only when this point is guaranteed.

1.4 How To Maintain Balance

Leafing through the biomechanical literature, two common definitions of stability are the following:

Structural Stability A walking system is structurally stable if and only if its center of mass projects vertically inside the convex hull of the contact points

Dynamic Stability A walking system is dynamically stable if and only if its ZMP projects vertically inside the convex hull of the contact points

The ZMP ("Zero Moment Point") is the point where the total reaction force of the ground does not produce any torque.

Unfortunately, as Wieber [10] pointed out, these criteria make sense only if all contact points between the robot and the environment lie on the same level. If the robot is climbing upstairs or is touching the environment with its hands, the above definitions are not anymore valid. We have reproduced Wieber's illustrative examples in Figure 1.4. Since one of our objectives is precisely to walk uphill, we cannot rely on these definitions in this project. Finding a universally valid definition of stability is still an open research area. Possibly, advances in this area might be crucial to improve the stability of biped robots, but for now we have to do without them.

Despite our understanding of the laws of physics, we do not know what makes an animal or robot maintain balance. However, several leading roboticians have made practical suggestions to improve a robot's balance. Let us briefly survey them:

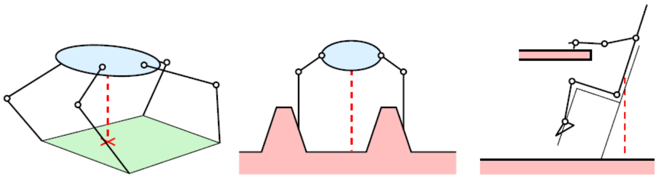


Figure 1: A walking system is generally supposed to be statically balanced if and only if its center of mass projects vertically inside the convex hull of its contact points (left). There are situations where it does but the system is not balanced (center), and situations where it doesn't but the system is balanced (right).

Ensuring that the entire foot touches the ground Vukobratovic [8] suggested that "dynamic balance is obtained by ensuring the whole foot's area, and not just the edges, is in contact with the ground"

Using small trunk motions The ASIMO team controls the stability of the robot (i.e. the position of its ZMP) by using small trunk motions (from the Sony website)

Adding reflexes According to Zehr and Stein [11], "reflexes function during human locomotion to preserve balance and ensure a stable walking pattern throughout the step." Taga [7] has suggested that "a mechanism of sensory-motor integration provides dynamical stability to human locomotion"

Synchronizing legs with arms McGeer [2] has explained that his Passive Dynamic Walker maintains balance in part because "arms are synchronized with the opposite leg during locomotion"

Sensing acceleration Most animals seem to be able to maintain balance in great part because they can sense acceleration, this sense being located in the inner ear

Designing improved leg trajectories Finally, several authors suggest that a key to improve stability may be to improve the leg's trajectories so that the torso is naturally steered, and one does not have to move the ankles to compensate its motion.

We may look back at these suggestions during this project to try to enhance our controller.

1.5 Designing Robot Controllers: Main Approaches

To end this brief introduction, we shall expose to the reader the main approaches that have been tried to design controllers for biped robots. These can be divided into three categories: ZMP based, passive dynamics, and CPG based.
The ZMP based approach is the one used by the HONDA team to delelop ASIMO, arguably the most successful robot developed until today. However, the ZMP approach relies on a pre-calculated trajectory, and, as a consequence, it cannot adapt its locomotion to new and unpredictable terrain conditions. This is the main drawback of this approach, and, since we are interested in designing an adaptive controller, it is not the approach chosen for this project. The ZMP, however, may remain a useful notion to analyze the performance of our controller.
Passive dynamics attempts to produce locomotion without active control, using solely the forces of nature (gravity and inertia). This line of research, pioneered by McGeer at MIT, is very interesting to uncover to what extent the forces of nature can be exploited to spare energy during locomotion. However, it is clear that animals *cannot* uniquely rely on the forces of nature to move. For example, when they move on sand or in water, animals obvioiusly must use active control. Most of the experiments that have been run using passive dynamics allow biped robots to move on a slope. In this project, we will not be concerned with this line of research.
The third class includes all attempts to replicate the key mechanisms that allow animals to move. It seems that a major part of a (vertebrate) animal motor control lies in the spinal cord, where small clusters of neurons, called "Central Pattern Generators" (CPGs) produce rhythmic signals that control the muscles used during locomotion. These CPGs can be modeled as systems of coupled non linear oscillators. According to Taga, including sensory feedback into a CPG based controller produces a stable locomotion pattern comparable to that seen in animals - a thesis that we will verify in the course of this project. Our robot controller (developed by Ludovic Righetti at BIRG under the supervision of Prof. Auke Ijspeert) follows this approach.

2 The Robot, and Its Controller

2.1 HOAP 2

In this project, we will develop a controller for a Fujitsu robot of the HOAP series. Here is a picture of a HOAP2 robot:

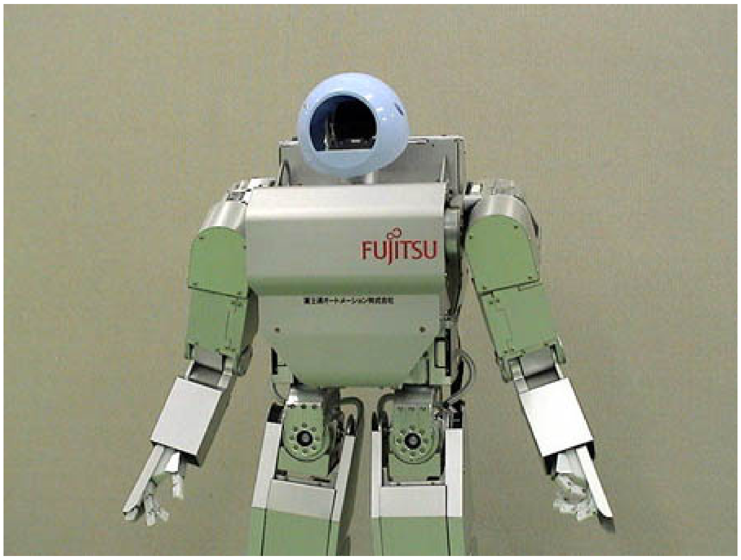


Figure 1: HOAP2 robot

Like all robots, a HOAP2 can be seen as a kinematic chain of links (arms, forearms, thighs, etc.) connected by joints (elbows, hips, knees, etc.). The joints of a HOAP2 are:

1. Waist (1 degree of freedom)
2. Legs (6 degrees of freedom x2)
3. Arms (4 degrees of freedom x2)
4. Hands (1 degree of freedom x2)
5. Neck (2 degrees of freedom)

This makes a total of 25 degrees of freedom (DOFs).
Currently, our controller only moves the legs of the robot, leaving waist, arms, neck, and hands fixed. The joints that we actively control are:

1. Hips (2 degrees of freedom x2)

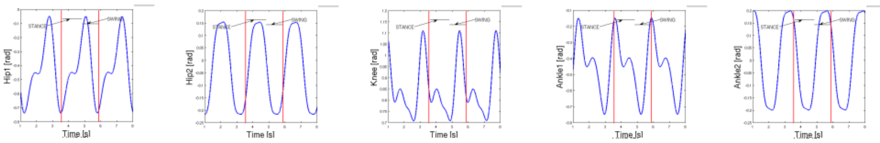
2. Knee (1 degree of freedom x2)
3. Ankles (2 degrees of freedom x2)

Figure 2: Controlled joints

That is, our controller controls 10 degrees of freedom out of 25.
The robot is only 50cm tall, and weights less than 7kg — which makes it suitable for human manipulation. Since the robot is very expensive, we will test our bio-inspired controller in simulation, preventing damages to the real robot. We will use Webots (Cyberbotics Ltd) as a simulation environment, and use the robot model developed by Pascal Cominoli, former EPFL student, in 2004/5.

2.2 Joint Trajectories

To produce locomotion, an animal must produce rhythmic signals to move its limbs. During walk, every degree of freedom follows its own trajectory.
The analysis of the DOF trajectories, called "gait analyisis", is an important tool to examine a robot's walk. This research field is also important to other disciplines, including the orthopedic sciences (to cure illnesses that impair walk, like Parkinson's disease, or tetraplegia), in sports training (to improve an athlete's performance), and in biometrics (to identify a person based on its gait). Gait analysis was born in the early XX century, thanks to the invention of photography that allowed to capture rapid sequences of movements during animal locomotion. The computer revolution in the mid XX century catalyzed research in the field, making more elaborated analyses of image sequences possible.
Our limb trajectories were provided by Fujitsu together with the HOAP2 robot. The trajectories are shown in Figure 3 (for the right leg). The interval between the two vertical lines (two consecutive heel strikes) corresponds to a step cycle. The corresponding step cycle is shown in Figure 4. The corresponding step cycle



hip1 trajectory hip2 trajectory knee trajectory ankle1 trajectory ankle2 trajectory

Figure 3: joint trajectories

is shown in Figure 4.

Figure 4: step cycle

All trajectories play a role during locomotion. The roles that each plays are known as the “determinants of gait”:

hip1 The hip1 trajectory produces the basic movement necessary for locomotion: the hips are flexed and extended rhythmically, so that the robot can make steps forward. If one uses only hips to walk, the resulting gait is known as “compass gait” — the most basic form of walking gait.

hip2 A compass gait is not very efficient, because the center of the pelvis rhythmically moves up and down (see Figure 5), a movement that wastes energy (to go up, one must fight against gravity) and makes balance difficult. The hip2 trajectory is useful to minimize the vertical motion of the pelvis.

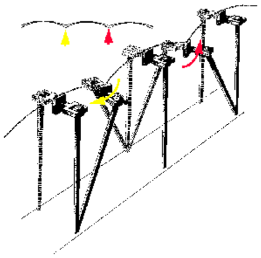


Figure 5: pelvis motion during compass gait

knee Unfortunately, solving one problem (the pelvis does not move up and down) creates another: now that the pelvis remains low, the legs tend to touch the ground during swing. To prevent this, the knee should be flexed during swing. Many gait pathologies (such as hemiplegia or diplegia) prevent the victim from bending the knee during swing, and causing compensatory movements like hip-hiking. A light flexion of the knee during stance is also advisable.

ankle1 The ankle is dorsiflexed at heel-strike and plantar-flexed at toe-off. This makes the step length a bit longer.

ankle2 The ankle2 trajectory compensates for the hip2 trajectory, so that the foot sole remains at all times parallel to the ground. In fact, the ankle2 trajectory is the same as the hip2 trajectory, but shifted by 180 degrees.

2.3 A Bio-Inspired Controller

Recent advances in neuroscience suggest that the DOF trajectories are generated in small clusters of neurons, called Central Pattern Generators (CPGs), distributed along an animal's spinal cord [1]. This explains why kittens whose spinal cord has been transected are still able to walk, or why decerebrated chickens can still run or fly short distances in the yard. While the main control mechanism happens in the spinal cord, higher control centers (mainly the brain stem, the cerebellum, and the motor cortex) play a role modulating these trajectories (to increase the speed of locomotion, take turns, etc.). Sensory feedback also plays a role in modulation, and is particularly important to improve balance.

How do CPGs produce rhythmic signals? Nature is full of rhythms. Hang a weight on the end of a rod, and you generate rhythmic behavior (a pendulum). Releasing the pendulum from a given height, and plotting the pendulum position over time, one obtains a periodic signal:

$$\theta(t) = A \cdot cos(\omega t) \tag{1}$$

where $\theta(t)$ is the angle of the pendulum, A the amplitude of oscillations, and ω the angular frequency.

If one starts the pendulum from a slightly greater height, one obtains again a periodic signal with the same angular frequency, but with a different amplitude. Instead of plotting the position over time, one can also depict the motion of the pendulum using a phase portrait — that is, showing the position and the velocity of the pendulum at any given moment. If one releases the pendulum from a certain height, the respective trajectory will be a closed curve. Releasing it from other heights, the trajectory will be another closed curve. One can see the signal, and its corresponding phase portrait, in Figure 6.

The rhythmic signals produced by Central Pattern Generators in the spinal cord tend to maintain the same frequency (just like a pendulum released from different heights), but also the same amplitude. Actually, most rhythmic signals produced by animals have these properties: the heart beat (if you make run the heart beat might change, but after a rest it becomes normal again), breath, the chewing system, etc. An oscillator that produces signals that tend to maintain both the same frequency and the same amplitude is slightly perturbed is called a "limit cycle oscillator"[6].

A simple limit cycle oscillator can be described with the equations:

$$\begin{cases} \dot{\rho} = \rho \cdot (1 - \rho^2) \\ \dot{\theta} = 1 \end{cases} \tag{2}$$

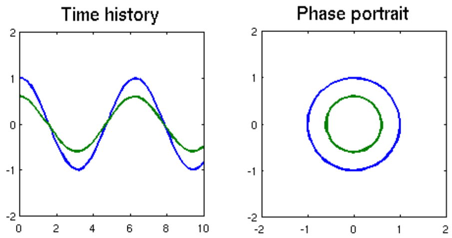


Figure 6: A linear oscillator

In Cartesian coordinates, this can be rewritten as:

$$\begin{cases} \dot{x} = (1-\rho^2)\cdot x - y \\ \dot{y} = (1-\rho^2)\cdot y + x \end{cases} \tag{3}$$

where $x = \rho \cdot cos\theta$ and $y = \rho \cdot sin\theta$ [5].
The signal that interests us is $x(t)$: the projection of the rotating point on the x coordinates. One can see its time series and its phase portrait in Figure 7.
Limit cycle oscillators are a good model to describe Central Pattern Generators.

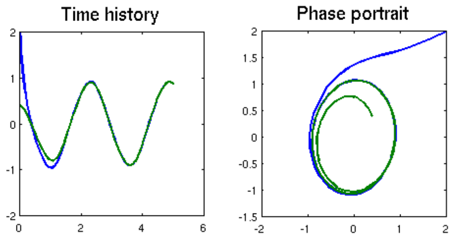


Figure 7: A limit cycle oscillator

Beside being a biologically inspired model, using limit cycles to generate DOF trajectories is useful because it allows to smoothly modulate, for example, the speed (frequency) of locomotion: by abruptly changing the radius of the limit cycle (the isolated, closed trajectory in phase space), the robot will converge to the new DOF trajectory in a smooth way. A simple command produces a smooth change. This mechanism is similar to what happens in vertebrate animals: the brain stem sends simple signals down the spine to modulate, for instance, the speed of locomotion.

2.4 A Mathematical Model

CPGs can be modelled using limit cycle oscillators, but the rhythmic signal does not have to be a simple sinusoid: as we have seen, the Fujitsu trajectories are more complicated than that. To reproduce the Fujitsu trajectories, one can add together various limit cycle oscillators, each one having a given frequency and a given amplitude — a sort of "Fourier series expansion" [3].
A general limit cycle oscillator can be described with the equations:

$$\begin{cases} \dot{x} = \gamma \cdot (\mu - \rho^2)x - \omega y \\ \dot{y} = \gamma \cdot (\mu - \rho^2)y + \omega x \end{cases} \tag{4}$$

where μ is the amplitude, ω the angular frequency, and γ the speed of convergence.
For a limit cycle oscillator to learn the (main) frequency of a teaching signal, one can use the following model (called "adaptive Hopf oscillator"):

$$\begin{cases} \dot{x} = \gamma \cdot (\mu - \rho^2)x - \omega y + \epsilon F(t) \\ \dot{y} = \gamma \cdot (\mu - \rho^2)y + \omega x \\ \dot{\omega} = -\epsilon F(t) \cdot y/\rho \end{cases} \tag{5}$$

where $F(t)$ is the teaching signal, and ϵ determines the learning rate.
To reproduce a Fujitsu trajectory, one can use a network of add together many adaptive Hopf oscillators (three are actually sufficient). For them to learn the frequency components of a Fujitsu trajectory, and its relative amplitude, Righetti has used the following model:

$$\begin{cases} \dot{x_i} = \gamma \cdot (\mu - \rho_i^2)x_i - \omega_i y_i + \epsilon A(t) \\ \dot{y_i} = \gamma \cdot (\mu - \rho_i^2)y_i + \omega_i x_i \\ \dot{\omega_i} = -\epsilon F(t) \cdot y_i/\rho_i \\ \dot{\alpha_i} = \eta x_i \cdot F(t) \\ A(t) = F(t) - \sum_{i=0}^{N} \alpha_i x_i \end{cases} \tag{6}$$

where i indicates the i^{th} oscillator, η and ϵ are parameters that control the learning rate, $A(t)$ is the signal that remains to be learned, and $F(t)$ is the Fujitsu trajectory.
The learned amplitudes are stored in α_i; the angular frequencies in ω_i. The

learned trajectory is:

$$F_{learned} = \sum_{i=0}^{N} \alpha_i x_i \tag{7}$$

It remains to ensure that the limit cycle oscillators maintain a desired phase shift with one another. For this, Righetti has modified the limit cycle oscillator as follows:

$$\begin{cases} \dot{x} = \gamma \cdot (\mu - \rho^2)x - \omega y + \epsilon F(t) + \tau sin(R_i) \\ \dot{y} = \gamma \cdot (\mu - \rho^2)y + \omega x \end{cases} \tag{8}$$

where τ is a coupling constant, and R_i is given by:

$$R_i = \frac{\omega_i}{\omega_0} sgn(x_0) cos^{-1} \left(-\frac{y_0}{\sqrt{x_0^2 + y_o^2}} \right) \tag{9}$$

Here, 0 indicates the limit cycle oscillator with lowest frequency.

Finally, one needs to learn the initial phase difference between the various oscillators — which can be achieved introducing the following equation:

$$\dot{\phi_i} = sin \left(R_i - sgn(x_i cos^{-1} \left(-\frac{y_i}{\sqrt{x_i^2 + y_i^2}} \right) \right) \tag{10}$$

We will not delve into further detail into this model. The interested reader can find further details in [3].

To produce locomotion, the DOF trajectories must be coordinated: a rabbit bounds by moving first its two front legs, and then its two back legs, with a phase shift of 0.5 between them; elephants amble moving their legs one after the other, with a phase shift of 0.25 between each; horses trot moving the front-right and rear-left legs in synchrony, and then the front-left and rear-right, with a 0.5 phase shift. Moreover, animals often change their gait by modifying the coordination pattern between different degrees of freedom, like horses switching from a trot to a gallop, or salamanders switching from a swim to a walk. We need to introduce this synchronization into the model.

Synchronization is a property that emerges when one couples together different limit cycle oscillators. For example, hanging two similar pendulums from the same wall is likely to result in them synchronizing their movements — a property first observed by Huygens in the seventeenth century. Nature is full of marvelous examples of similar synchronization: crickets chirp in unison; fireflies flash in synchrony; pacemaker cells in the heart fire maintaining a constant phase relation [6].

To produce synchronization between the DOF trajectories, one needs to couple the limit cycle oscillators. A well known coupling scheme that produces collective synchronization is Kuramoto's scheme:

$$\dot{\theta}_i = \omega_i + K/N \sum_{j=1}^{N} sin(\theta_j - \theta_i), i = 1, ..., N \tag{11}$$

Righetti has adapted this scheme to produce the desired phase shift between the joint trajectories. The equations used are:

$$\dot{x}_{(0,k)} = (\mu - \rho^2) \cdot x_{(0,k)} - \omega_{(0,k)} \cdot y_{(0,k)} + \tau sin(\theta_{0,k} - \phi_{0,k}) \tag{12}$$

$$\dot{\phi}_{(0,k)} = sin(\theta_{(0,k-1)} - \theta_{(0,k)} - \phi_{(0,k)}) \tag{13}$$

where (0,k) is the first oscillator of the kth CPG.

Once again, we will not delve any further into the details of this model. The interested reader can find the details in [4].

We can now reproduce the Fujitsu trajectories using limit cycle oscillators to move our robot. Animals convert the DOF trajectories into limb movements using biological molecular motors that use the energy stored in the bonds of ATP. Our robot, in contrast, converts the DOF trajectories into mechanical motion using an electric motor mounted on its back. The development of synthetic molecular motors, which are more efficient than electric motors, is still an active research area in nanotechnology. Currently, they have severe limitations that need to be overcome — but they might become a reality in the near future.

2.5 The Robot Sensors

To gather information from the environment, animals use senses. Beside the five senses of sight, hearing, touch, smell, and taste (these are the five senses first identified by Aristotle), animals rely on others to survive. Birds can detect changes in the magnetic field to find their way back home after a migration; some fishes detect changes in the electric field to communicate; bats (but also dolphins and whales) utilize a biological sonar to locate objects. A sense that plays a major role in animal locomotion is the sense of balance, located in the inner ear. A HOAP2 robot is endowed with three senses: the sense of balance, the sense of touch, and the sense of sight (optional).

The sense of balance is particularly important for locomotion. It allows cats, for example, to walk on a thin fence. It is located in the inner ear, and is determined by the level of a fluid (the "endolymph") in the labyrinth. Our robot is endowed with an equivalent sensor that allows him to measure the tilt of the body: a gyroscope. Gyroscopes are basically spinning wheels that, due to the conservation of angular momentum, tend to maintain the same orientation in space. When the robot bends on one side, its (sagittal) axis forms an angle with the axis of the gyroscope. Measuring this angle, the robot can measure the tilt of the body.

Another important sense used in animal locomotion is sight, which allows the animal to know if there are obstacles on its way, if it is walking up- or downhill, if there is a ravine just before him. Walking with the eyes closed is difficult, and if the animal does not know its environment well, it must move with great attention because it might just fall down a cliff at any moment. A HOAP2 robot has a camera to enable it to see the surroundings. However, since it is still very difficult to design good algorithms to interpret the images captured by the camera, we cannot rely yet on vision to produce stable locomotion. For the moment, we will not even use the cameras. Our robot must try to walk with its eyes closed.

A third sense that is believed to be important for animals to walk is the sense of touch: pressoreceptors located under the foot soles allow to detect the reaction forces of the ground during locomotion. They are important to detect the body sway, but also to adapt the rhythm of walking depending on the timing of a foot landing on the ground. Moreover, animals seem to have built-in responses to similar stimuli that play a role in behavior. For example, if an animal (unexpectedly) feels there is no ground under his foot, it may extend the leg to look for contact. The role of reflexes in animal locomotion is receiving greater attention by the research community.

Using only the Fujitsu trajectories, with no sensory feedback, the robot falls after 12 seconds. We want to improve this record. This is going to be our job in the rest of this project. Possibly, the robot should be able to speed up, slow down, walk backwards, walk up- or down-hill, and the like. One can see the robot walk with no sensory feedback in $oldWalk.mpeg$.

3 Adding Sensory Feedback

In this section, we wish to design reflexes to improve the robot's walk.
As in [9], reflexes will be added to the output of the Fujitsu trajectories as follows:

$$\theta(t) = \alpha(t) + \beta(t) \tag{14}$$

where $\theta(t)$ is the input command to a joint angle, $\alpha(t)$ is the Fujitsu trajectory, and $\beta(t)$ is the reflex.
Possibly, $\beta(t)$ should be a smooth function (i.e., a function that is infinitely differentiable). This ensures that the robot can actually move the joints as desired, and prevents damages to the motor. Before looking for useful reflexes, we need to endow the robot with the sense of touch.

3.1 The Sense of Touch

To endow the robot with foot sensors, we have used the Open Dynamic Engine, a C++ library useful to simulate rigid body dynamics. Luckily, Yariv Bachar, a researcher at the university of Edinburgh, borrowed us a copy of the code he wrote to implement the foot sensors on a HOAP2 robot, which we only had to modify to fit our needs. We will not bore the reader with the details of the code here. If he is interested, he can look it up in Appendix I.
When a foot strikes the ground, two forces act on the robot: a normal force N (that allows the robot to remain *on* the ground), and a frictional force F. Friction prevents animals from slipping and, thus, allows them to move around. The two forces are related according to Coulomb's approximation: $F = \mu \cdot N$. Here, μ is the frictional coefficient, which depends on the ground surface (in our simulation: 0.995), as well as on the foot's surface. Coulomb's law can be seen at work when one holds a glass of water: if one does not exert sufficient pressure with the fingers (the *normal* force), the glass falls down (the *frictional* force is insufficient to balance gravity). The two forces are shown in Figure 8.

Figure 8: normal force and frictional force

Our foot sensors detect the *normal* force exerted by the ground on the feet. Each foot is endowed with 4 foot sensors, as shown in Figure 9. The footprints were left by the robot walking *on* the page.

In our simulation, the eight sensors are located in the only points where the foot might touch the ground. Hence, if one adds up the values measured by the foot sensors, one should obtain the total normal force exerted by the ground on the robot. When the robot is static, this force should equal the robot's weight times the acceleration of gravity. This is a good occasion to test whether our foot sensors work. The robot weights 7 kilograms, and the acceleration of mass is 9.81 m/s^2, and, hence, adding up the sensor values one should obtain little less than 70 Newton. Let us see if this is true.

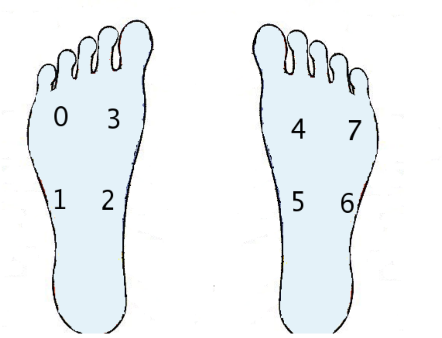


Figure 9: foot sensors

In Figure 10, the robot stands on his right foot. The sensor values are: $S_4 = 19$; $S_5 = 13$; $S_6 = 15$; and $S_7 = 21$. Adding them up, one obtains a total force of 68 (Newton).

In Figure 11, the robot bends forward. The sensor values are: $S_0 sensor0 = 23$; $S_1 = 2$; $S_2 = 11$; and $S_3 = 32$. Adding up the values, one obtains 68 (Newton).

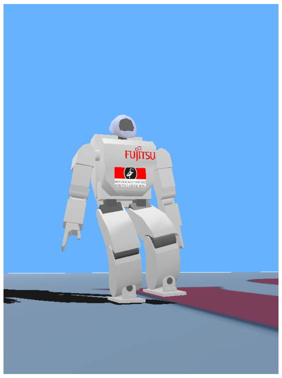


Figure 10: sensor test (1)

One can try to put the robot in all sorts of static positions. The resulting force always adds up to about 70 (Newton). The sensor values make sense.
In our simulation, the robot controller and the physics library are two distinct programs, running in parallel. Every time the robot's feet impact the ground, the physics library sends the measured values to the controller, that can use them to introduce sensory feedback into the system.
The foot sensors work. Now, we can add reflexes to improve the robot's walk.

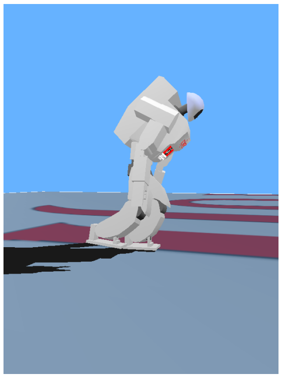


Figure 11: sensor test (2)

3.2 Methodology

Before designing reflexes, we should ask: is it possible to predict what reflexes will help stability, prior to running simulations? Our guess is that this is hard. Take a bicycle. Up until the mid 1960s, scientists believed that a bike maintained balance because of the gyroscopic effects of the wheels. In 1970, however, David Jones built a now famous "unridable bike" that showed that a bike's stability is not due to gyroscopic effects. So, we know a bicycle maintains balance, but we do not know why. It is likely to we can improve the balance of our robot, but probably we will not fully understand why — and certainly, we will not be able to predict this in advance. For curiosity, we show to the reader Jones' unridable bike in Figure 12.

Another possibility to motivate the design of a reflex is to have a look at what

Figure 12: Jones' bike

reflexes animals use to move in the world. However, even here a note of caution is mandatory: our robot is different from real animals in important ways: it has no toes, it has rectangular feet, and is made of hard material. Is it likely that reflexes that are useful for an animal with little feet, toes and soft skin are also useful for our robot? Possibly, but it is a mistake to assume that it must be this way: our robot looks more like a human wearing ski shoes than a human walking bare foot. Hence, we will consider the inference "if reflex A is useful in the animal kingdom, then it is useful also for biped robots" not valid.

How else can we motivate the design of reflexes? Sincerely, I have no idea. Maybe, the only viable solution is to try any reflex loop, run simulations, and see if it works. This does not sound very scientific, but it is the only solution that comes to our mind. We will play dice.

3.3 An Ankle Reflex

Vukobratovic [8], arguably the most famous robotician in the world, suggests that the key ingredient to improve a robot's balance is to adjust the ankles so that the contact area between foot and ground is maximized:

> Dynamic balance is obtained by ensuring the whole foot's area, and not just the edges, is in contact with the ground.

At first sight, it seems that Vukobratovic' advice is plainly wrong: at the end of the stance phase, it is helpful to use the tip of the foot to take longer steps, and if the robot is falling down on one side, it should exert a force on the *edge* of the foot to remain upright. Furthermore, one can observe that this advice is not usually obeyed in the animal kingdom: cats and dogs, for example, walk on their toes to walk faster; horses even walk on the *tips* of their toes. However, let us assume that, in general, Vukobratovic' advice is good (we will actually see that it is *very* good). Let us come up with a reflex that goes in this direction.

The simplest reflex is:

$$\beta(t) = B \cdot c(t) \tag{15}$$

B is a parameter, and $c(t)$ is the foot's center of pressure (the sum of the sensor values on the right of the sole, minus the sum of the sensor values on the left). Let us see, running a few simulations, whether this simple reflex is of any help. With a pleasant surprise, we found that even such a basic reflex loop is *very* helpful. Here are the results of some simulations, modifying the parameter B:

Parameter B	walk time [sec]
1/500	2
1/1000	38
1/5000	54
1/7000	∞
1/10000	∞
1/20000	32
1/100000	20
0	12

With a wide range of parameter values (from about 1/100000 to 1/1000!) the simple ankle reflex seems to help. With $1/6000 < B < 1/15000$, the robot even manages to walk on and on, without ever (ever, for us, means at least one minute) falling.

Let us see how the reflex affects the ankle2 trajectory. The new ankle2 trajectory with $B = 1/10000$ in shown in Figure 13 (left). The jigglings under the curve indicate the presence of a negative feedback loop: the reflex keeps correcting the ankle. On the right, we have showed the difference between the trajectory without (red) and with (blue) ankle reflex. The difference is of the order of 0.015 radians.

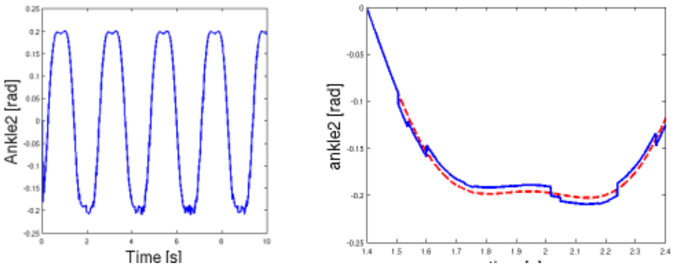


Figure 13: new ankle2 trajectory (desired)

In Figure 14 we have shown the corresponding *real* ankle2 trajectory (a robot cannot always move its limbs as desired). Here, the jigglings are absent. On the

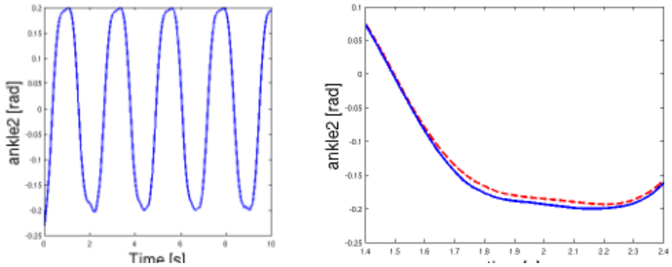


Figure 14: new ankle2 trajectory (real)

right, we have shown the difference between the old (red) and new (blue) real trajectories. One way to see if this reflex maximizes the contact area between foot and ground is to plot the evolution of the center of pressure during stance. The center of pressure oscillates, as shown in Figure 17 (left). These oscillations are caused by the jigglings.

There is a huge literature on how to get rid of oscillations in a negative feedback loop. Solutions are usually *ad hoc*, and include using a PID controller, filters, and fuzzy logic. After a few unsuccessful attempts, we have modified the reflex as follows:

$$\beta(t) = B_2 \cdot c(t)/(t - t_0) \tag{16}$$

where B_2 is a parameter, and t_0 is the last time the foot landed.

The idea is that the corrections to the ankle become smaller and smaller during stance. This solution proved fine. In Figure 18, one can see the new ankle2 trajectory. The jigglings have disappeared. In Figure 19 we have shown the corresponding *real* trajectory. We can plot again the evolution of the foot's center of pressure during stance. It is shown in Figure 17 (right). This time, the oscillations are clearly damped.

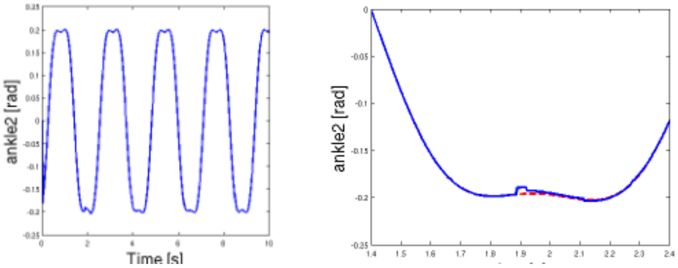


Figure 15: new ankle2 desired trajectory (bis)

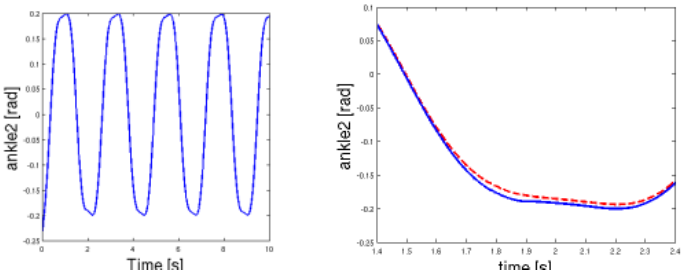


Figure 16: new ankle2 real trajectory (bis)

Another way to compare the two is by computing the variance of the center of pressure. With the new reflex, this has dropped significantly from $V_1 \simeq 1600$ to

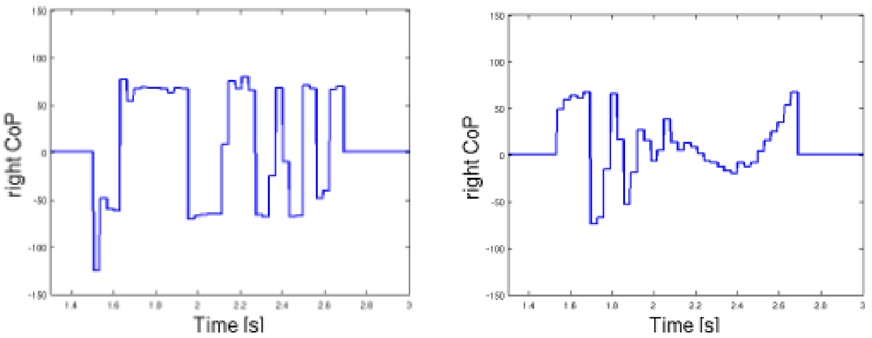


Figure 17: foot's center of pressure during stance

$V_2 \simeq 800$.
Running simulations, the new ankle control seemed to provide good results: the robot walks on and on without falling, and the walk looked more stable.
We have not been completely faithful to our original promises, so far: we said that we wanted our reflexes to be smooth functions of time, but up to now they are not. So, let us try now to turn the reflex into a differential equation, and see if it is still useful:

$$\dot{\beta} = B_3 \cdot \dot{c}(t) \tag{17}$$

Once again, running simulations we had a good surprise: the robot walk was stable with a wide range of parameters. Here are the results of the simulations:

Parameter B_3	walk time [sec]
10	1
5	1
2	∞
1	∞
1/5	∞
1/10	∞
1/20	∞
1/30	∞
1/40	∞
1/1000	∞
1/10000	19
0	13

The robot walks on and on, as long as parameter B_3 remains within the interval $[1/1000; 2]$.
In Figure 18 we have shown how the new reflex affects the ankle2 desired trajectory with $B_3 = 1/30$. In Figure 19 we have shown the corresponding *real* trajectory.
This is the best result so far, and we will keep it in our following simulations.

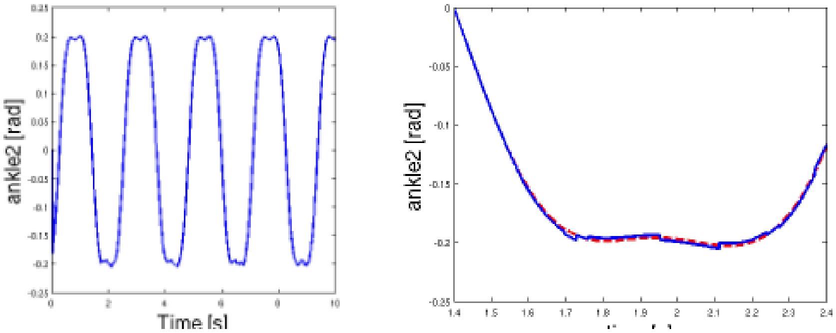


Figure 18: new ankle2 desired trajectory (tris)

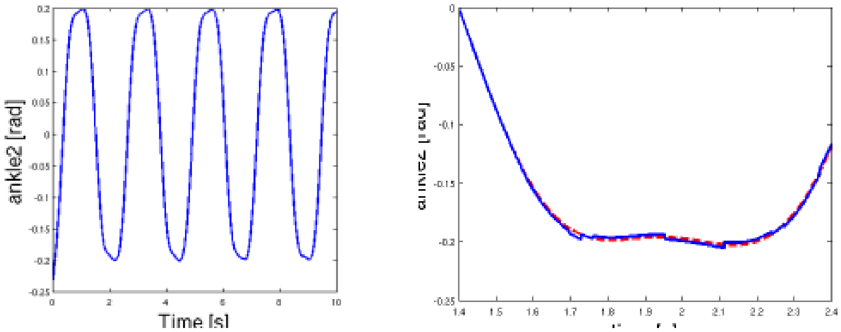


Figure 19: new ankle2 real trajectory (tris)

3.4 A Knee Reflex

Observing the robot's walk, one can notice that the walk does not look very nice: the robot always oscillates back and forth while standing on its left foot — which does not happen when its stands on its right foot. It would be nice if we could get rid of these oscillations. To get rid of these oscillations, one can try to do all sorts of things: land with a flat foot, walk with a higher frequency, etc. After some fruitless attempts, we have found a solution that seemed to provide good results: bending the knees while landing.

The knee reflex works as follows: as soon as a single support phase begins, the robot bends its landing knee to soften the landing. Due to this knee flexion, the robot foot impacts the ground with a greater contact area — which is in line with Vukobratovic's advice.

This knee reflex can be modelled with the equation:

$$\beta(t) = B \cdot (H(t - t_0) - H(t - t_0 - \epsilon)) \tag{18}$$

where B is a parameter, t_0 the reflex activation time, ϵ the duration of the reflex, and $H(x)$ a Heaviside function defined as:

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x > 0 \end{cases} \tag{19}$$

This function is not differentiable. As a consequence, one may use the following approximation:

$$\dot{\beta} = \begin{cases} B & t_0 < t < t_0 + \delta \\ -B & t_0 + \delta + \epsilon > t > t_0 + 2 \cdot \delta + \epsilon \\ 0 & otherwise \end{cases} \tag{20}$$

where B is a parameter, t_0 the reflex activation time, δ a short interval of time, and ϵ the duration of the reflex.
This function is differentiable, but not *infinitely* differentiable. One might also use the following smooth approximation:

$$\beta(t) = B \cdot \left(\left(\frac{1}{(1 + e^{-2k(t-t_0-\delta)}}) \right) - \left(\frac{1}{(1 + e^{-2k(t-t_0-\delta-\epsilon)}}) \right) \right) \tag{21}$$

where B is a parameter, t_0 the reflex activation time, δ a short interval of time, and ϵ the duration of the reflex. The only condition to use this equation is that k and δ must be chosen so that $\beta(t_0) \simeq 0$.
For convenience, in our simulations we will use equation 20, which provided us with good experimental results. The control principle is shown in Figure 20.

The important parameter that allows to modify the reflex is ϵ (the duration), as illustrated in Figure 21.

Figure 20: the knee reflex

Let us see what happens if we modify the duration, with B=10 and $\delta = 0.1$. First, we experiment using only the knee reflex, and getting rid of the ankle reflex. Here are some results:

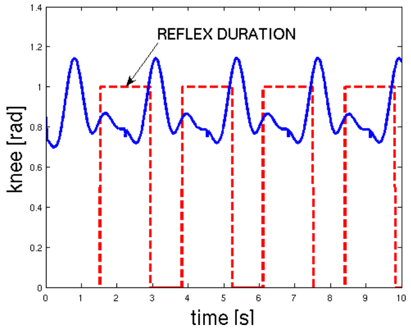


Figure 21: knee control

Duration of reflex [ms]	walk time [sec]
700	17
600	40
500	15
400	31
300	38
200	21
100	19
50	17
10	17
1	13

The knee reflex is useful with $300 < B_{knee} < 600$, but not as much as the ankle reflex. In Figure 22 we have shown how the knee reflex affects the knee desired trajectory with $B = 300$. In Figure 22 we have shown the corresponding *real* trajectory.

A combination of the ankle and the knee reflexes provide the best results in our

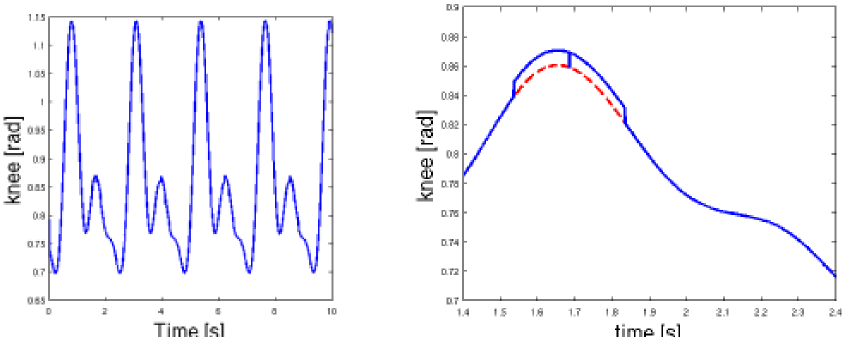


Figure 22: new knee trajectory (desired)

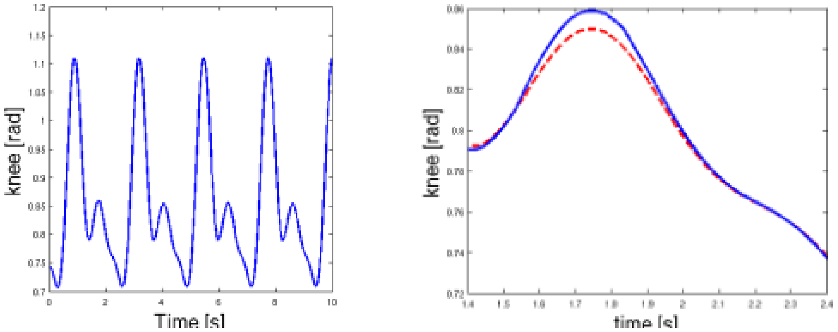


Figure 23: new knee trajectory (real)

simulations. We will have the occasion to verify how the two reflexes combine in more difficult conditions (increasing speed, etc.) because the robot already walks well using only the ankle reflex.

3.5 A Hip Reflex

We might be satisfied with our ankle and knee reflexes: the robot walks on and on with a wide range of parameters. However, before developing the touch sensors, Righetti had written a reflex triggered by the gyroscope that affects the hip2 trajectory. We should check whether it is still useful, or we can do without it.

The gyroscope allows to measure the tilt of the robot in the frontal and in the sagittal plane. The two anatomical planes, often used in the robotics literature, are shown in Figure 24.

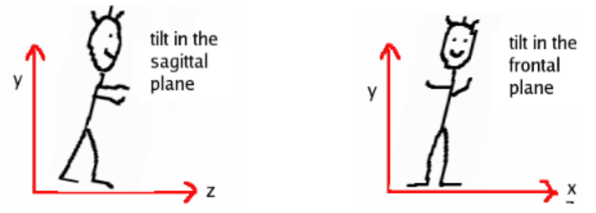


Figure 24: sagittal (left) and frontal (right) anatomical planes

Righetti designed the reflex to minimize lateral tilt: when the robot is tilted on one side, the robot compensates this tilt using his hip2 joint. The equation used to correct the robot's tilt is:

$$G(t) = B \cdot \alpha(t) \tag{22}$$

where G is the gain of the hip, k is a constant, and α is the value measured by the gyroscope (in radians).

The gain was added to the radius of the limit cycle describing the hip2 joint, as follows:

$$\dot{x} = ((1 + G(t)/\rho) - \rho^2) \cdot x - \omega y \tag{23}$$

$$\dot{y} = ((1 + G(t)/\rho) - \rho^2) \cdot y + \omega x \tag{24}$$

Since a lateral tilt of the hip affects the angle between the foot and the ground, the hip2 reflex was accompanied by a reflex that tilts the corresponding ankle2 joint in the opposite direction. The control principle is shown in Figure 25.

We have run simulations modifying parameter B_1, while keeping $B_2 = 100$.

Figure 25: the hip2 reflex

Here is a table showing some results:

Parameter B_1	walk time [sec]
1000	5
2010	22
2020	18
2040	31
2060	20
2080	22
2100	20
2500	19
2700	17
3000	9

For a wide range of parameters, from about 1500 to 2700, the hip control is useful.

One can see the effect of the feedback loop on the hip2 desired trajectory with $B_1 = 2040$ and $B_2 = 100$ in Figure 26. The corresponding *real* trajectory is shown in Figure 27. In theory, we should see similar (opposite) changes in the

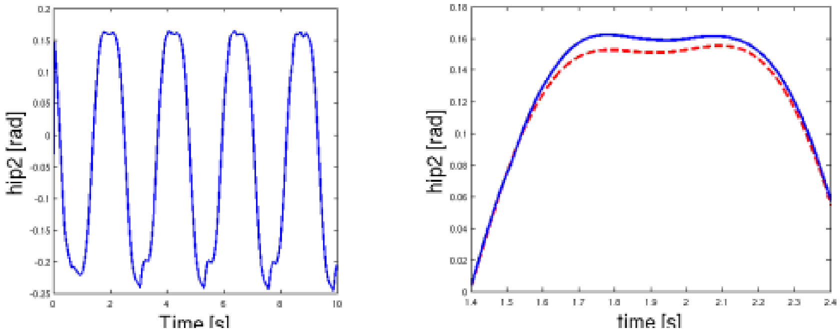


Figure 26: new hip2 trajectory (desired)

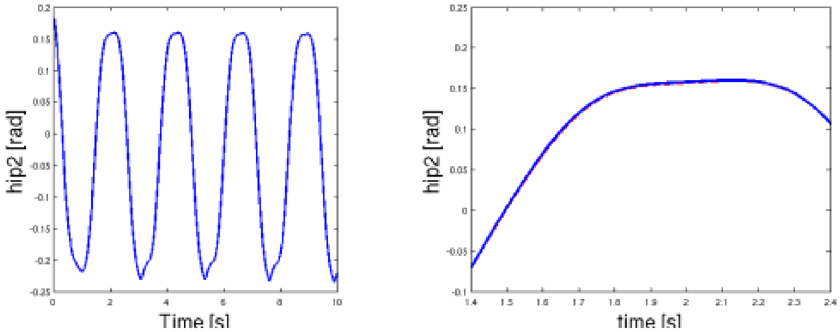


Figure 27: new hip2 trajectory (real)

ankle2 trajectories. Let us see if it is the case. In Figure 28 we have shown the ankle2 desired trajectory. In Figure 29 we have shown the corresponding *real* trajectory.

The trajectories look indeed specular. We may mention that this reflex, that adjusts the tilt of the robot sensing the inclination of the body, resembles the vestibular (or labyrinthine) reflex used by animals.

The hip reflex is useful for a wide range of parameters. In the following table we summarize the results of our simulations. A '1' indicates that a given reflex was used, a '0' indicates otherwise.

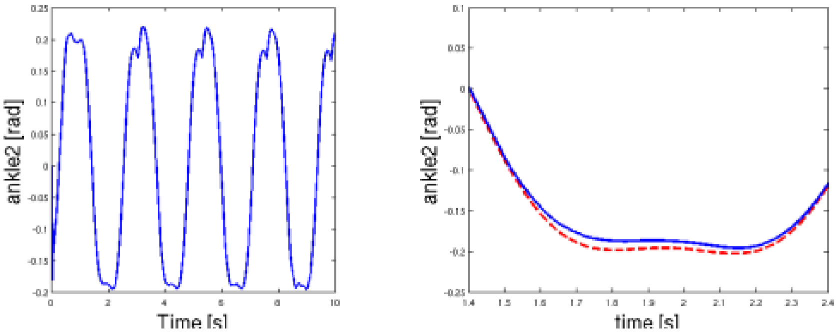


Figure 28: ankle2 desired trajectory

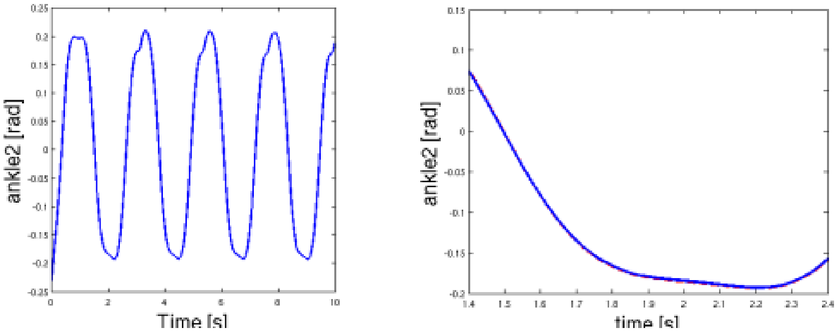


Figure 29: ankle2 real trajectory

Hip Control	Ankle Control	Knee Control	Walk Duration [sec]
0	0	0	12
1	0	0	31
0	1	0	∞
0	0	1	40
1	1	0	39
1	0	1	24
0	1	1	∞
1	1	1	∞

All reflexes are useful. The ankle reflex is from far the most important one, and the knee reflex seems to further improve the robot's walk. The hip reflex provided good results, but using it in conjunction with the other reflexes actually worsened the performance of the robot. As a consequence, in the following simulations we will keep the ankle and the knee reflexes, and we will not use the hip reflex anymore.

Now that the robot walks without falling on a flat surface, we can increase the difficulties. In the next section, we will try to speed up, slow down, walk backwards, walk up-hill, and the like.

4 Advanced Walking

Now that we managed to walk without falling on an even surface, we can attempt to increase the difficulties: walk faster, walk up-hill, walk up-stairs, etc.

4.1 Speeding Up

To start with, let us try to speed up. In our previous simulations we used the original Fujitsu trajectories, increasing their frequency by 10 percent. Now, we want to go faster.

Let us increase the frequency by 20 percent. Using no reflexes, the robot falls after 5 seconds. Let us see how the ankle reflex can help:

B_{Ankle}	walk time [sec]
1	1
1/5	26
1/10	27
1/20	21
1/30	13

This time, the ankle reflex is not sufficient, and the parameter space that provides a satisfactory walk has considerably shrinked (the interval is now [1/5; 1/10]). Let us see if the knee reflex helps:

B_{Knee}	walk time [sec]
600	5
400	10
300	9
200	5
100	5

The robot performace is terrible.

Then, we can combine the ankle and the knee reflexes. Keeping $B_{Ankle} = 1/5$, let us modify B_{Knee}:

B_{Knee}	B_{Ankle}	walk time [sec]
600	1/5	∞
400	1/5	∞
300	1/5	∞
200	1/5	∞
100	1/5	∞

What a good surprise! While the ankle reflex alone and the knee reflex alone provided poor results, both of them together give excellent results! The ankle and the knee reflex work well together.

Let us increase the frequency by 30 percent. Keeping the optimal parameter settings just used, the robot falls after 15 seconds. Let us see if we can improve the walk modifying the ankle parameter:

B_{Ankle}	walk time [sec]
1	1
1/3	20
1/5	15

Playing with the ankle parameter is not sufficient. Second, we can play with the duration of the knee reflex (keeping B=1/3):

B_{Knee}	walk time [sec]
0	16
50	20
100	25
150	20
1000	14

This time, not even the knee reflex proves sufficient. Since the robot keeps falling forward or backwards (but not laterally), we can try to improve the walk by adding an ankle control on the *sagittal* plane. The equation governing the new reflex is:

$$\dot{\beta} = B_4 \cdot \dot{c}(t) \tag{25}$$

where $c(t)$ this time is the evolution of the center of mass along the sagittal plane.

Here are the results of the simulations with the addition of the new ankle control:

$B_{Ankle-bis}$	B_{Ankle}	B_{Knee}	walk time [sec]
1/3	3/2	200	15
1/3	1	150	15
1/3	1/3	0	28
1/3	1/3	100	45
1/3	1/3	150	∞
1/3	1	200	∞
1/3	1/2	200	∞
1/2	1/3	200	∞
1/3	1/3	200	∞
1/3	1/3	250	∞
1/3	1/3	300	∞
1/3	1/3	400	∞
1/3	1/3	500	∞
1/4	1/3	200	∞
1/5	1/3	200	35
1/10	1/3	200	37
1/3	1/3	600	11
0	1/3	200	25

A great improvement! The ankle control in the sagittal plane proves crucial to maintain balance at this speed. Now the robot walks on and on without losing stability as long as $B_{Ankle-bis}$ remains in the interval [1/4; 1], B_{Ankle} in the interval [1/3; 1], and B_{Knee} in [150; 500].

Since we have added a new reflex, we should briefly show how it modifies the ankle1 trajectory, just like we did for the other reflexes. In Figure 30 one can see the new desired ankle1 trajectory, and the difference between the new and the old ones. In Figure 31 one can see the corresponding *real* trajectory.

Let us increase the frequency by 40 percent. Keeping the parameters used for a 30 percent increase of speed (with $B_{Ankle-bis} = 1/4$), the robot falls after 21 seconds. That's not too bad, but we may wish to do better. Let us see more precisely how the various parameters affect the walk. Using only the ankle con-

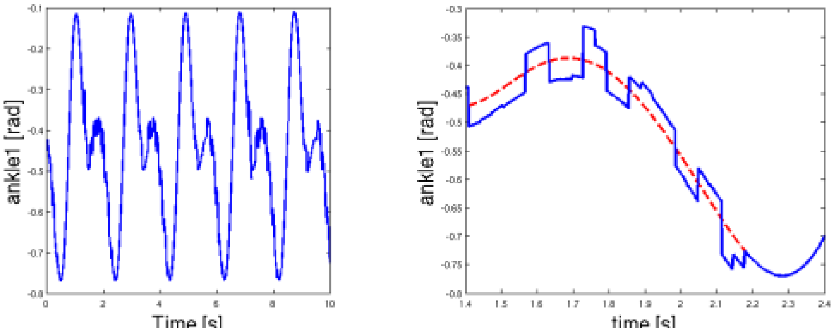


Figure 30: ankle1 desired trajectory

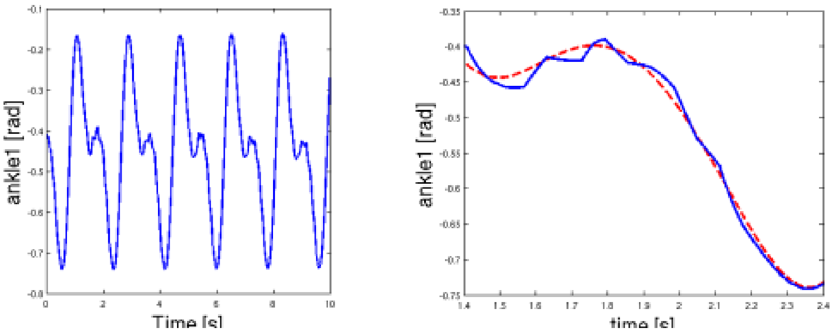


Figure 31: ankle1 real trajectory

trol, we have obtained the following results:

B_{Ankle}	walk time [sec]
1	6
1/2	7
1/3	6
1/5	4

Using only the ankle control provides a very poor performance. Let us try to add the knee reflex:

B_{Knee}	B_{Ankle}	walk time [sec]
25	1/2	4
100	1/2	6
200	1/2	12
250	1/2	13
300	1/2	4

Let us try to add the sagittal ankle control. Here are some results:

B_{Knee}	B_{Ankle}	$B_{Ankle-bis}$	walk time [sec]
200	1/2	1/3	17
200	1/2	2/5	∞
200	1/2	1/2	22

The optimal performance happens with $B_{Ankle-bis} = 2/5$. Once again, the sagittal ankle reflex appears to be crucial to attain “high” speeds.

Like before, this last reflex proves to be the most important to improve the walk. In our simulation, the robot walks during 34 seconds (which is not too bad), and then falls.

Let us increase the frequency by 50 percent. Keeping the optimal parameter settings just used, the robot walk lasts 15 seconds. This is not too bad. Improving significantly this record, however, proved difficult. We should set this as the upper limit we can reach.

We can summarize what we have found out: as speed is increased, the ankle controls must be strenghtened, and the knee reflex duration must be shortened. This is a qualitative finding. We may wish to provide a more quantitative scheme. Here is a scheme that fits well with the results of our simulations:

Parameter	Speed - Parameter Relationship
$B_{Ankle-bis}$	As speed is multiplied by x, $B_{Ankle-bis}$ is multiplied by x^2
B_{Ankle}	As speed is multiplied by x, B_{Ankle} is multiplied by x^3
B_{Knee}	As speed is multiplied by x, B_{Knee} is raised to the power $1/x^2$

This scheme works within the limits of speed the robot can attain (that is, up to a 40 percent increase of speed). We shall remind the reader that, at normal speed, $B_{Ankle-bis}$ and B_{Ankle} are set to 1/10, and B_{nee} is set to 500.

One can see a movie of the robot walking with a speed increased by 30 percent in $newWalk1.mpeg$.

4.2 Turning

Now that we can modulate our speed of locomotion, we want to learn to turn. To turn, we will need to use a DOF that we have not used yet: the rotation of the robot's hips. To start with, we can try to do the following (to turn right): while the robot stands on its right foot, the right hip rotates to the left by an angle α, and while the robot stands on its left foot, the right hip rotates to the right by an angle β. To do this, we can synchronize the right hip3 trajectory with the left hip1 trajectory: when the robot projects forward its left leg, it also rotates its right hip to the left. Running simulations with $\alpha = 0.13$ and $\beta = 0.11$, the results are very promising: the robot turns left with ease, with a radius of about 0.65 meters. To run the simulations, we have used a speed increased by 30 percent.

Let us see how the hip3 trajectory is modified. In Figure 32 one can see the old hip3 desired trajectory (which is obviously flat, since the robot did not use its hips). In Figure 33 one can see the new hip3 trajectory for the right leg, during a turn.

We can modify the parameters α and β to see how this affects the turn radius. We will keep the quotient α/β constant. Here are the results of the simulations:

$\alpha + \beta$ [rad]	turn radius [m]
0.02	1.17
0.07	0.85
0.15	0.7
0.17	0.65

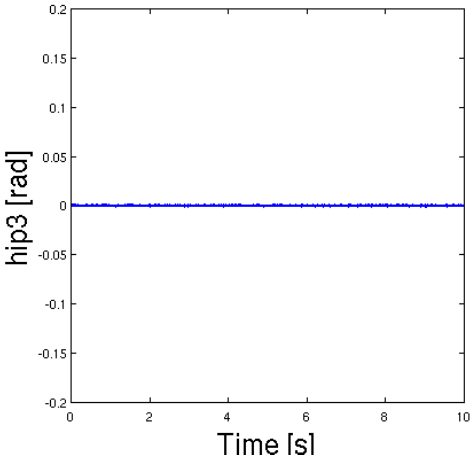


Figure 32: hip3 desired trajectory (old)

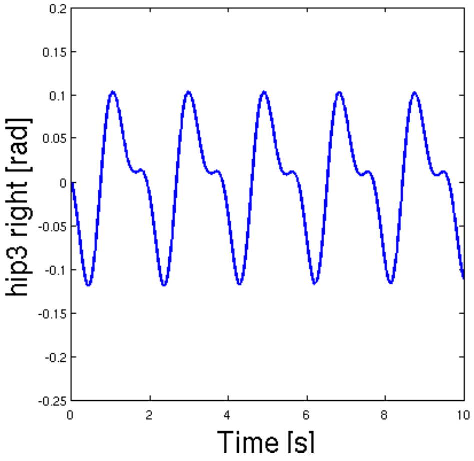


Figure 33: hip3 right desired trajectory (new)

The smallest turn radius we could reach is 0.65 meters. With smaller radiuses, the robot falls before completing a full circle.
One can see a movie of the robot turning right with $\alpha+\beta = 0.17$ in $newWalk2.mpeg$.

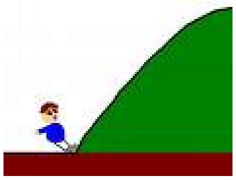
4.3 Walking Backwards

Now that we can speed up and turn, we wish to walk backwards. This should not be difficult, in theory: it suffices to invert all the DOF trajectories. One way to do this is to invert the walking frequency in our controller. Let us see if

the robot can walk backwards.
The results of the simulations are good: the robot walks backwards with ease. Apparently, it even has a greater facility walking backwards than forward. One can keep the reflexes used for forward locomotion during backwards locomotion. One can see a movie of the robot walk backwards in $newWalk3.mpeg$.

4.4 Walking On a Slope

Next, we wish our robot to walk up slopes. To simulate a hill, instead of bending the terrain on which the robot is walking, we have bent the acceleration of gravity.



To walk up slopes, we have designed a new reflex using the DOF trajectory of the robot's torso, which we had never used before. The reflex is triggered by changes in the values of the gyroscope, and is governed by the following equation:

$$\dot{\beta}(t) = B \cdot \dot{c}(t) \tag{26}$$

where c(t) is the value returned by the gyroscope, and B a parameter that we have set to 800.
With these parameter settings, we can walk up a 1 degree and a 2 degree slopes with ease, but we get stuck with a 3 degree slope. To walk up a 3 degree slope, we can set the reflex parameters as follows:

B_{Knee}	B_{Ankle}	$B_{Ankle-bis}$	B_{torso}	walk time [sec]
800	0.8	0.4	800	∞

Let us see how the new reflex affects the body joint trajectory. In Figure 34 one can see the old body desired trajectory (which is, again, flat). In Figure 35 one can see the new body trajectory. Let us try to walk up a 4 degrees slope. To walk up this slope, we needed to adapt the parameters as follows:

B_{Knee}	B_{Ankle}	$B_{Ankle-bis}$	B_{torso}	amplitude	walk time [sec]
800	0.8	0.4	900	0.76	∞

We can also manage to walk up a 5 degree slope. However, for this we have tu tune the parameters very precisely:

B_{Knee}	B_{Ankle}	$B_{Ankle-bis}$	B_{torso}	amplitude	frequency	walk time [sec]
800	0.8	0.4	960	0.72	1.4	32

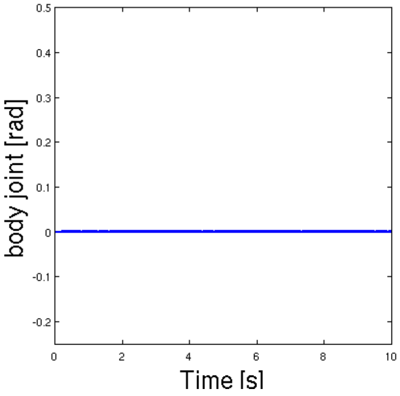


Figure 34: body joint desired trajectory (old)

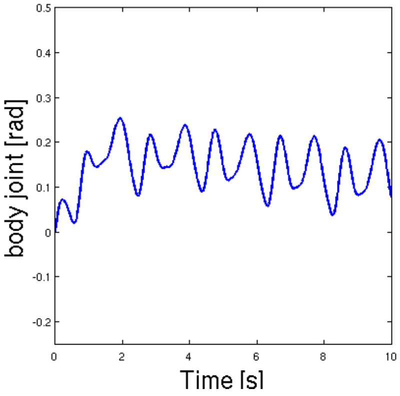


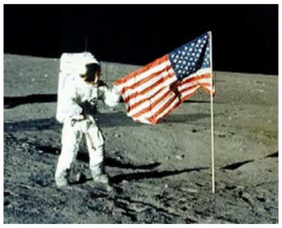
Figure 35: body joint desired trajectory (new)

With these parameter settings, the robot manages to climb up 5 degree slopes. With steeper hills, he walks up with less ease.

One can see a movie of the robot walk on a 4 degree slope in *uphill.mpeg*.

4.5 Walking With Reduced Gravity

Research on legged robot locomotion is important to explore terrains inaccessible to humans, like distant planets. Locomotion on distant planets is an interesting research topic because of the changes in gravity. For example, gravity on the moon is about 6 times weaker than on earth. According to scientists, walk on the moon cannot be as fast as on earth because it involves a continuous



transformation of potential energy into kinetic energy, and the available energy is reduced. In this section, we wish to verify how our robot can cope with a reduced gravity.

To start with, let us see how the walking frequency must be modified with various gravitational accelerations. According to the theory of ballistic walk, if gravity is reduced by a factor of x, then the walking frequency should be reduced by a factor of $\sqrt{x}$ because the swing time of a pendulum is inversely proportional to the square root of gravity. As a consequence, if an animal is walking on the moon (where gravity is 1 sixth that on the earth), it should walk at a frequency reduced by a factor of $\sqrt{6} = 2.45$.

Running some experiments with reduced gravity, we have obtained the following results:

gravitational acceleration	best walking frequency [sec]
-9.81	1.3
-7	1.1
-6	1.0
-5	0.9
-4	0.85
-3.5	0.8
-3.3	0.75
-3	0.7
-2.5	0.65
-2	0.59
-1.635	0.53

These results confirm the theory of ballistic walk.

5 Conclusions

5.1 Project Objectives

The objective of this project was to improve a bio-inspired robot controller adding reflexes so that he could walk, speed up, slow down, walk up-hill, down-hill, and take turns.

At the start, I was not sure the objective was attainable. I was asked to improve the robot controller adding reflexes - but are reflexes really the key ingredient to improve balance? I felt like I was asked to prepare the best cappuccino finding the optimal amount of sugar - but what about cream, coffee, or chocolate? What if the key ingredient was to add a sensor, modify the CPG equations, use another coupling scheme, use other reference trajectories, add toes, compute the ZMP, or add chewingums under the feet? Eventually, I found simple reflexes that greatly improved the robot's balance. Reflexes turned out to be a key ingredient. A happy ending.

During the project, a fair amount of time was spent investigating the other ingredients that allow to prepare a working, bio-inspired robot controller: what sensors are important, how one can model an animal's CPGs, why using Hopf oscillators, why the reference trajectories look the way they do, how can one ensure balance, etc. We felt that these questions were important to a better understanding of the field, and to justify our design choices. Moreover, since research into bio-inspired robotics is profitable not only to build robots, but also to better understand animals, having a look at the literature in gait analysis, prosthetic science, or sports science seemed in order.

For what concerns the project objective, the robot manages to walk, speed up, slow down, climb up-hill, down-hill, and take turns without losing balance, as long as he does not have to walk at huge speeds, climb up vertical walls, or walk down a ravine. We have shown in a scheme how the parameters controlling the reflexes used should be adapted to suit different walk conditions. One can see the robot performance in the movies on the website birg.epfl.ch.

5.2 Personal Knowledge

It is common usage in the write-up of theses to mention not only the objectives attained, but also the personal knowledge acquired.

The master project was a good occasion to improve my knowledge of:

1. Linux
2. Webots
3. C++
4. Matlab

5. LaTeX
6. Gimp

Moreover, the project was a good occasion to read the literature in animal locomotion, bio-inspired robotics, neuroscience, gait analysis, sports science, and prosthetics. It improved my knowledge of an animal's motor system, the role of the various limb trajectories in animal locomotion, the role of reflexes in animal behavior, the interplay of the animal senses to maintain balance, and the like. In this sense, I think the project has been interesting and enriching.

5.3 Problems Encountered

Most problems encountered during the project were: am I doing the right thing? One problem, as mentioned, was whether it was better to try to design reflexes to improve the robot's walk, or look for other ingredients. Why not use the ZMP? Why not add a sensor? Why not try to understand why the trajectories provided by Fujitsu look this way, or try to use others? Why not read the literature in prosthetics, or try to understand what trajectories are considered optimal in the sports sciences? Why use the Kuramoto coupling scheme, and why use Hopf oscillators? There were plenty of topics that called for investigation, and I was always wondering: should I try to design reflexes, or look for other ingredients to improve the robot's stability?

Another problem, as mentioned, was the lack of a justification for designing a reflex. It appeared difficult to justify a reflex starting with Newton's laws of physics, and designing reflexes inspired by the animal kingdom did not appear to be a convincing solution, either. How to justify a given design, then? It seemed that the only viable method was to try any reflex loop (say, if sensor x is active, then the robot should move joint y), run simulations, modify the parameter settings, and see if the robot's walk improved. But is this a really a good way to proceed? I have not found any alternative. Maybe, this was really the only way.

5.4 Limitations

Among the limitations of this project we may mention:

1. we have not found a way to justify the design of a reflex before running simulations. Possibly, there might be good arguments to predict *a priori* what reflexes will be useful, but we have found none;
2. we have designed a few reflexes that proved useful to improve the robot's walk. However, there may be a thousands of others that we have not tested at all, and that may work better;
3. all reflexes designed are triggered by touch sensor or gyroscope values. Apparently, animals heavily rely on vision to maintain balance. We have

not implemented vision on the robot, and, hence, no reflex is activated by visual inputs.

4. of the 25 robot degrees of freedom, we only used those in the hips, knees, ankles, and torso. One might try to use other degrees of freedom to improve stability (for example, synchronizing the arms with the legs)

5. we have only worked in simulation. It is not clear that all the designed reflexes would work on the real robot, due to, for instance, noise.

5.5 Future Directions

The field of bio-inspired robotics is still in its infancy, and there are plenty of areas to investigate. Let us mention a few directions that seem promising:

1. Developing vision (or other senses) for the robot
2. Learning from experience
3. Making short term predictions
4. Expressing personality traits
5. Playing soccer

References

[1] S. Grillner. Neural networks for vertebrate locomotion. *Scientific American*, pages 48–53, January 1996.

[2] T. McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82, 1990.

[3] L. Righetti, J. Buchli, and A.J. Ijspeert. From dynamic hebbian learning for oscillators to adaptive central pattern generators. In *Proceedings of 3rd International Symposium on Adaptive Motion in Animals and Machines – AMAM 2005*. Verlag ISLE, Ilmenau, 2005. Full paper on CD.

[4] L. Righetti and A.J. Ijspeert. Programmable central pattern generators: an application to biped locomotion control. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 2006.

[5] S. Strogatz. *Nonlinear Dynamics and Chaos. With applications to Physics, Biology, Chemistry, and Engineering.* Addison Wesley Publishing Company, 1994.

[6] S.H. Strogatz and I. Stewart. Coupled oscillators and biological synchronization. *Scientific american*, 269(6):68–74, 1993.

[7] G. Taga. A model of the neuro-musculo-skeletal system for human locomotion. I. Emergence of basic gait. *Biological Cybernetics*, 73(2):97–111, 1995.

[8] M. Vukobratovic and B. Borovac. Zero-moment point – thirty five years of its life. *International Journal of Humanoid Robots*, 1(1):157–173, 2004.

[9] Gunag Wang, Qiang Huang, Juhong Geng, Hongbin Deng, and Kejie Li. Cooperation of dynamic patterns and sensory reflex for humanoid walking. In *ICRA*, pages 2472–2477. IEEE, 2003.

[10] P. B. Wieber. On the stability of walking systems. In *Proceedings of the International Workshop on Humanoid and Human Friendly Robotics*, 2002.

[11] P. Zehr and R. Stein. What functions do reflexes serve during human locomotion? *Progress in Neurophysiology*, 58(1):185–205, 1999.