A biped walking machine

Table of Contents:

1 Introduction

In this introduction a short historical overview over the development of biped walking machines will be given. It includes the scientific motivations that lead scientists to build humanoid robots and introduces some of the most sophisticated biped robots today. Last but not least, the concepts of walking will be briefly presented at the end of the introduction.

- Purposes
  One of the main purposes of a biped machine is to be able to walk autonomously. Compared to the NASA rover-robot a biped could easily step over stones or jump over a hole in the ground and would not have to plan an additional itinerary for every hindrance object. The reason the NASA did not make use of a biped in its mission until now is very simple. Even though there exists a considerable number of two-legged robots today the issue of creating a biped machine that has the same stability control and physical abilities like a human being is far from being resolved. Stability control and the generation of energy-optimized walking trajectories are still one of the main research issues in the field of humanoid robots. This is why speaking of a biped walking machine means addressing all of the following topics: Sensing, Actuating, Planing and Control.
  An interesting fact is that a jogging robot consumes about 10 times as much energy as its human counterpart, the power/weight ratio of the machine being very bad. This illustrates quite well why a biped robot today is not yet able to perform the same physical tasks as a human being.
  The main aims in the field of humanoid robotics are the following:
  1. Understanding the bio mechanics of the human body, that means its structure and its behavior.
  2. Understanding of the human cognition processes. This field of study comprises the questions of how we learn from sensory information and how we acquire perceptual motor skill. An aim is to build computational models of these abilities.
  3. Building better orthosis and prosthesis was one of the initial aims of the research in humanoid robots. Some examples are: powered leg prosthesis for neuromuscularly impaired, ankle-foot othosis, biological realistic leg prosthesis and forearm prosthesis.
  4. Performance of human mental and physical tasks. It is expected that humanoid robots will be able in the near future of performing tasks like personal assistance, interacting in a social environment with humans or performing dirty or dangerous jobs.
  5. Entertainment.

- Historical Overview
  1495: A first humanoid automaton was designed by Leonardo da Vinci approximately about the year 1495. It is the first known human design for a robot. The automaton is a knight which is able to make some human like motions like sitting up and moving its arms and neck.
  1738: Jacques de Vaucanson builds The Flute Player, a life-size figure of a shepherd that could play the flute and the Tambourine Player.
  1921: The Czech writer Karel Capek introduced the word *robot* in his play *R.U.R.* The word comes from *robota* in czech what means forced labor and drudgery (slavery work).
  1970: A Serb engineer *Miomir Vukobratovic* introduced a theoretical model to explain biped locomotion based on the *Zero Moment Point.* Vukobratovic was one of the pioneers in humanoid robotics.

1973: The so called robot *Wabot-1* was build at Waseda University in Tokyo. This humanoid-like machine was capable of communicating in Japanese and to measure distances using external receptors.
1986: HONDA initiated its research and development program in humanoid robotics that will result in the following years in the conception of several biped robots. The desired goal was to develop a robot able to coexist and collaborate with humans and to perform tasks impossible or to dangerous for us. The proclaimed aim was to create a mobile robot which brings additional value to human society. Thus the creation of a robot able to meet consumer needs, and not a robot that is limited to specialized operations.
1995: The robot *Wabian* was build at Waseda University in Tokyo. It had a total of 35 mechanical dof and was able to walk forward and backward and to carry load. It was the work of *Ichiro Kato*, considered today as the father of Japanese robotic research.
2000: HONDA creates its 11th bipedal humanoid robot, ASIMO.
2001: Sony unveils another very sophisticated humanoid robot, called *Sony Dream Robot*. It will be renamed QRIO in 2003.

- Sophisticated robot platforms today: ASIMO, HRP-2 and JOHNNIE
  - → ASIMO
    The name is an acronym for "**A**dvanced **S**tep in **I**nnovative **MO**bility". In Japanese it is pronounced *Ashimo* what, not coincidentally, means "legs also". It is probably the most sophisticated humanoid robot today. It' s height is about 1m30 and the weight is 54 kilograms (batteries included). It has about 34 mechanical dof and officially can run at a speed of about 3 km/h, though HONDA announced last year that it had reached a running speed of 6 km/h. ASIMO mainly distinguishes itself by its features of recognition technology and its capability to interact with humans on a simple level. It is able to recognize dynamic objects and to interpret postures and gestures. Thus it can react to natural movements of human beings, for example greeting a person approaching him (or it). Another interesting feature is its capability of face recognition. It also can distinguish different sounds and identify their direction.
  - → HRP-2
    HRP-2 of Kawada Industries is the final robotic platform of the *Humanoid Robotics Project* initiated by the *Manufacturing Science and Technology Center* in Japan. Its specifications are a height of about 1m54 and a weight of 58 kg (batteries included). It has 30 dof and a walking speed of about 2 km/h. An interesting feature is the cantilevered crotch joint that improves walking performance in a confined, uneven area. Cooling systems incorporated in the leg actuators allow enhanced continuous stepping performance.
    The striking external appearance was designed by Mr. Yutaka Izubuchi, a mechanical animation designer famous for his robots appearing in Japanese anime.
  - → JOHNNIE
    Johnnie was designed in the framework of the DFG *Priority Program Autonomous Walking*. The main objective was to realize an autonomous 2-legged walking machine with dynamically stable gait patterns, able to walk on uneven surfaces and around curves. JOHNNIE has a total of 17 joints, its weight is about 40 kg and its height is 1m80. Its walking speed is approximately 2km/h. Each leg incorporates 6 joints: three hip joints, one joint actuating the knee and two ankle joints (pitch and roll). The upper body has a rotational dof about the vertical body axis. The joints are driven by DC Brush motors in combination with light weight gears. Incremental encoders are used in each joint to measure joint angles and velocities. Two 6 axis force sensors are integrated in the feet to measure ground reaction forces. Additionally there is an orientation sensor (3 axis accelerometer and 3 gyros sensors) to evaluate the spacial orientation of the upper body. The control algorithms run on a PC on a RT-operation system (RT-Linux). The function of gait generation and control will be discussed later.

- Principles of biped locomotion
  The particular motion and stability control of the human being is based on the interaction between the nervous system and the human body. An interesting fact is that the capability of performing rhythmical and synchronized motions of the legs is acquired by a baby a few days after its birth.
  The following figure illustrates the interaction of the nervous system, the cognitive sensors and the human body. The nervous system might be compared to a controller interpreting the cognitive signals acquired from the environment in order to realize a stable dynamic walking motion. The following figure illustrates this:

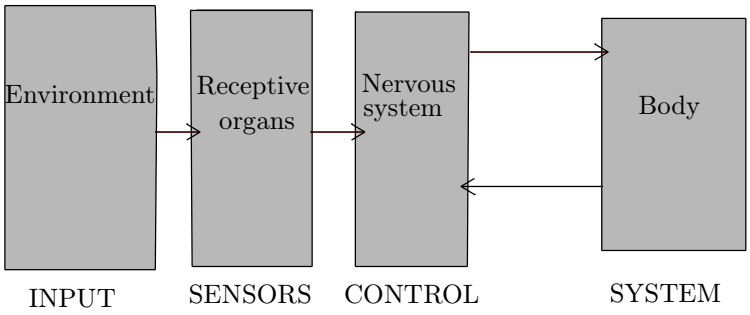

figure1.1

(source: 6)

An important concept of biped locomotion is the Zero Moment Point. The contact polygon area between the ground and the feet is shown in the following figure.

stable region

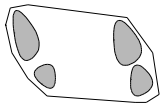

figure1.2

(source: 6)

The Zero Moment Point is defined as the point on the contact surface, where the resultant moments of all contact forces are zero. As the contact forces are due to gravitation and inertia of the walking body, the ZMP can also be defined as the point on the surface where the moment of the resultant inertia forces (the combination of inertia and gravity forces) becomes zero.
Static and dynamic walking have to be discussed. Static walking is defined as a motion where the projected CoM is always inside the contact polygon, whereas in the case of dynamic walking the CoM is not always above the stable region(see above). Thus dynamic walking produces short intervals of instability. In Order to analyze dynamic walking, we must formulate constraints on the position of the ZMP. This will be done later.

2 Important Concepts in Robotics

In this section some important concepts in robotics will be presented. In the perspective of properly modeling and simulating any kind of robot, it is important to understand a few basics rules and ideas that apply not only to industrial robots, but also to biped machines. This is why they are introduced in this chapter. The concepts presented are the computation of forward and backward kinematics based on the *Denavit-Hartenberg* convention of attaching frames to the links of a robotic structure and the *Jacobian* Matrix used for transformations between *Work* and *Configuration Space*.

- Denavit-Hartenberg Notation
  Mechanical design considerations favor joints with one degree of freedom. There exists *revolute* and *prismatic* joints. If a joint has several degrees of freedom, it still can be modeled as a series of revolute and prismatic ones. This is why we can concentrate on the representation of 1-dof joints connecting the different links of the manipulator. A link is considered as a *rigid body* defining the relationship of two neighboring joint axis.
  In order to compute the kinematics of a robotic structure and to define the location of each link relative to its neighbors, we define a frame attached to every link of the manipulator. The convention of affixing frames to links that leads to the *Denavit-Hartenberg notation* is presented here. The following rules apply:

  1. The Z-axis of frame i is coincident with the joint axis i.
  2. The a_i perpendicular is defined as the mutual perpendicular between joint axis i and i+1. It can be interpreted as the distance between the corresponding joint axes.
  3. The origin of frame i is located where the a_i perpendicular intersects with joint axis i.
  4. X_i points along a_i in the direction from joint i to joint i+1

  If the link frames have been attached according to this convention, the following definitions of link parameters are valid:

  - a_i = the distance from Z_i to Z_{i+1} measured along X_i
  - α_i = the angle from Z_i to Z_{i+1} measured about X_i
  - d_i = the distance from X_{i-1} to X_i measured along Z_i
  - θ_i = the angle from X_{i-1} to X_i measured about Z_i

  We chose $a_i > 0$, because it corresponds to a distance. The other link parameters are signed quantities. In the case of a revolute joint, θ is called the *joint variable*, whereas α, a and d are fixed *link parameters*. In the case of a prismatic joint the joint variable is d. These 4 parameters are sufficient to describe the link itself and its connection to neighboring links. The described relations are illustrated in the figure below:

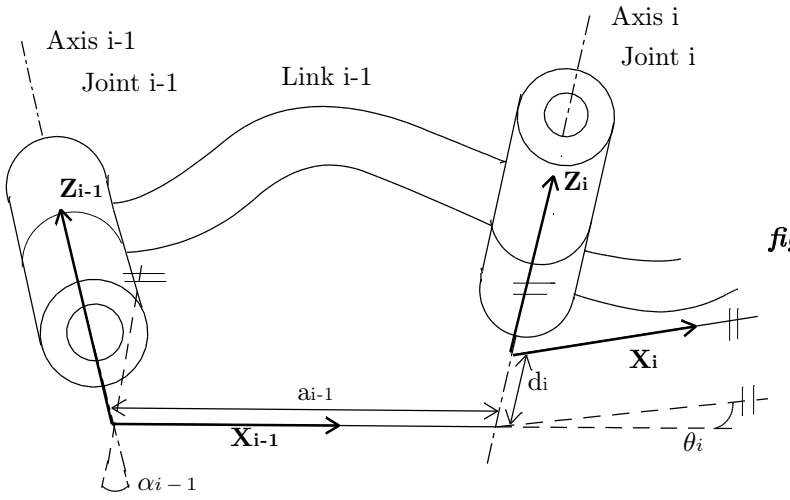

figure2.1

(source: 1)

a, d, α and θ are known as the Denavit-Hartenberg parameters. Now we can write the transform that defines frame {i} relative to frame {i-1}. The transformation is a function of the one joint variable defined above. It can be written as follows:

(2.1) $T_{i-1i} = T_{i-1\,R}\, T_{\mathrm{RQ}}\, T_{\mathrm{QP}}\, T_{\mathrm{Pi}}$

R, Q, and P are intermediate frames that are obtained by:

1. P := Translation of frame {i} along the $Z_i -$ axis by the amount of $- d_i$
2. Q := Rotation of frame {P} about the Z_i-axis by the amount of $-\theta_i$
3. R := Translation of frame {Q} along the X_{i-1}-axis by the amount of $-a_{i-1}$
4. i-1 := Rotation of frame {R} about the X_{i-1}-axis by the amount of $-\alpha_{i-1}$

Thus T_{i-1i} can be written as:

(2.2) $T_{i-1i} = R_x(-\alpha_{i-1})\, D_x(a_{i-1})\, R_z(-\theta_i)\, D_z(d_i)$

The positive signs of D_z and D_x must be interpreted carefully. They refer to a point being translated relative to a reference frame. As a forward-translation of a point equals a backward-translation of the reference frame, we assign a positive value to the corresponding translation coordinate in the transformation matrix.
In order to realize a transformation that includes both, rotation and translation, we use a 4x4 Matrix:

(2.3) $T_{\mathrm{AB}} = \begin{pmatrix} R_{\mathrm{AB}} & P_{\mathrm{ABorg}} \\ 0\,0\,0 & 1 \end{pmatrix}$

with R_{AB} being the 3x3 Rotation Matrix and P_{ABorg} the point B in coordinates of frame {A}.
For illustration the transformations R_x(-α_{i-1}) and $D_x(a_{i-1})$ are given below. R_z and D_z are equivalent.

(2.4) $R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & -s\alpha & 0 \\ 0 & s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad D_x = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ with $a = a_{i-1}$ and $\alpha = \alpha_{i-1}$

Multiplying out (2.2), we obtain the following general form of $T_{i-1\,i}$:

$$(2.5)\quad T_{i-1\,i} = \begin{pmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

with $\sin\alpha = s\alpha$ and $\cos\alpha = c\alpha$

(2.5) defines the relationship between frame {i-1} and frame {i}

Instead of supplying more definitions and explanations, a short example will suffice to illustrate the practical use of the Denavit-Hartenberg notation:
The following manipulator is given:

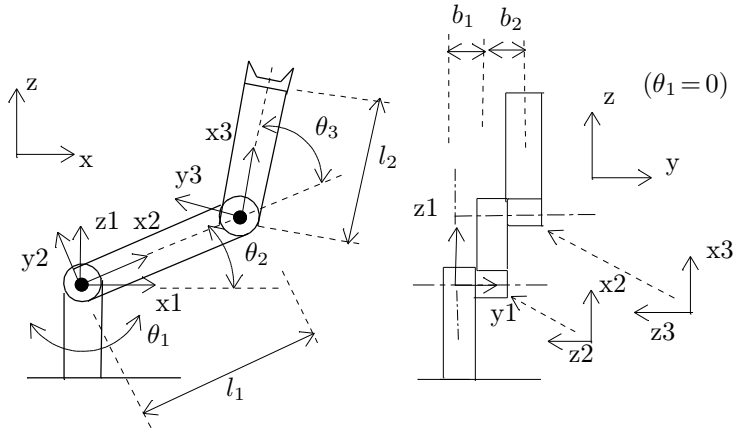

figure2.2

The frames are attached according to the rules mentioned above. The first joint has 2 dof but is modeled as two successive 1dof rotational joints. The manipulator might be thought of as a leg of a biped robot, joints 1+2 actuating the hip and joint 3 the knee. The base might be thought of as the trunk of the robot. We obtain the following Denavit-Hartenberg parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	$\pi/2$	0	-b_1	θ_2
3	0	l_1	-b_2	θ_3

and transformation definitions (2.5):

$$(2.6)\, T_{01} = \begin{pmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.7) T_{12} = \begin{pmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & -1 & b_1 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.8) T_{23} = \begin{pmatrix} c\theta_3 & -s\theta_3 & 0 & l_1 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & -b_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(2.6) to (2.8) describes the transformations between the successive frames. The multiplication yields:

$$(2.9)\quad T_{01}\, T_{12}\, T_{23} =$$

$$T_{03} = \begin{pmatrix} \frac{1}{2}c_{3-1+2} + \frac{1}{2}c_{1+2+3} & -\frac{1}{2}s_{3-1+2} - \frac{1}{2}s_{1+2+3} & s_1 & \frac{1}{2}l_1 c_{-1+2} + \frac{1}{2}l_1 c_{1+2} - s_1 b_2 - s_1 b_1 \\ \frac{1}{2}s_{1+2+3} - \frac{1}{2}s_{3-1+2} & \frac{1}{2}c_{1+2+3} - \frac{1}{2}c_{3-1+2} & -c_1 & \frac{1}{2}l_1 s_{1+2} - \frac{1}{2}l_1 s_{2-1} + c_1 b_2 + c_1 b_1 \\ s_{2+3} & c_{2+3} & 0 & s_2 l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(2.9) gives us the possibility of computing the absolute position of a point P defined in frame {3} as a function of the joint variables θ_1, θ_2 and θ_3. Thus if we define P_{feet_3} as $(\ l_2\ \ 0\ \ 0\ \ 1\ )^\top$ in coordinates of frame {3}, we get the position of the foot relative to the trunk of the robot by premultiplying P_{feet_3} with T_{03}.

(2.10) $\quad P_{\text{feet}_o} = T_{03} P_{\text{feet}_3}$

- Forward and inverse kinematics
  The example above serves as an illustration for *forward kinematics*. If we know the actual configuration of the robot, we can calculate the position of the robot's arms or legs as a function of its joint variables θ_i (in the case of revolute joints). The vector θ_i constitutes the *configuration space* of the robot.
  The more difficult converse problem focuses on the following question: given the desired position and orientation of a link at the end of a kinematic chain, how can we compute the set of joint angles which will achieve the desired result? The solution to such an *inverse kinematic* problem can be very difficult. In the case of an industrial robot the design of the robot is aimed to facilitate the solution of the inverse kinematic problem. A very simple example of a planar biped shall serve as an illustration.

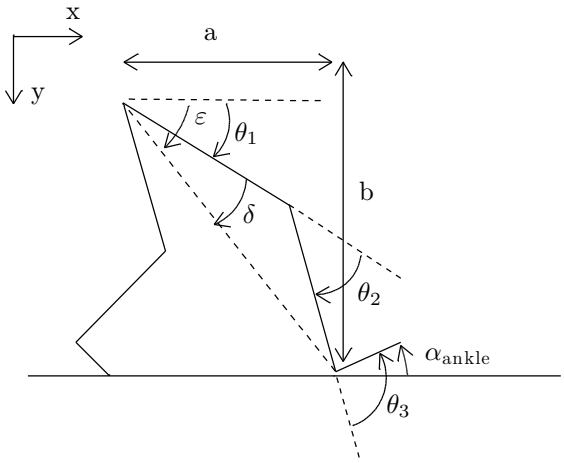

figure2.3

Given is the position of the ankle relative to the pelvis expressed in coordinates a, b and the orientation of the foot of the robot relative to the ground (α_{ankle}). We are looking for the joint values that fulfill the given conditions. The length of the upper leg shall be l_1 and the length of the lower leg is given by l_2. The problem can be solved geometrically:
First we check whether the given problem has a solution by verifying the following condition

(2.11) $\quad \sqrt{a^2+b^2} \le l_1 + l_2$

If a solution exists we obtain θ_2 by applying the law of cosines, we use the simplification $\cos(\pi+\theta_2) = -\cos\theta_2$ and obtain (2.13):

(2.12) $a^2+b^2 = l_1^2 + l_2^2 + 2l_1 l_2 \cos\theta_2$ $\quad$ (2.13) $\quad \theta_2 = \text{acos}(\frac{a^2+b^2-(l_1^2+l_2^2)}{2l_1 l_2})$

Next we calculate ε and δ (law of cosines):

(2.14) $\varepsilon = \text{atan}(\frac{b}{a})$ $\qquad$ (2.15) $\delta = \text{acos}(\frac{l_1^2+a^2+b^2-l_2^2}{l_1\sqrt{a^2+b^2}})$

Now we can obtain θ_1 and θ_3:

(2.16) $\quad \theta_1 = \varepsilon - \delta$ $\qquad$ (2.17) $\quad \theta_3 = -(\theta_1+\theta_2+\alpha_{\text{ankle}})$ $\quad$ (with $\alpha_{\text{ankle}} > 0$)

- Jacobian Matrix
  The *Jacobian* is a multidimensional form of the derivative. If we got a vector of m functions, each function having n independent variables

  $$y_1 = f_1(x_1, x_2, ..., x_n) \tag{2.17}$$

  $$...$$

  $$y_m = f_m(x_1, x_2, ...x_n)$$

  we can calculate the differentials of y_i and obtain

  $$\delta y_1 = \frac{\delta f_1}{\delta x_1}\delta x_1 + ... + \frac{\delta f_1}{\delta x_n}\delta x_n \tag{2.18}$$

  $$...$$

  $$\delta y_m = \frac{\delta f_m}{\delta x_1}\delta x_1 + ... + \frac{\delta f_m}{\delta x_n}\delta x_n$$

  which can be written in vector notation:

  $$\delta Y = \frac{\partial F}{d\partial X}\delta X \quad \text{or} \quad \delta Y = J(X)\delta X \tag{2.19, 2.20}$$

  J(X) is a m x n matrix consisting of the partial derivatives of Y and is called the Jacobian. By dividing both sides of (2.20) by a differential time element we can think of the Jacobian as mapping velocities in X to those in Y:

  $$\delta Y^{\cdot} = J(X)\delta X^{\cdot} \tag{2.21}$$

  J(X) is a linear transformation and is time-variant, thus it is called a time-varying linear transformation.
  In the case of robotics Jacobians are used to map joint velocities to Cartesian velocities of one particular joint. To compute the Jacobian, we must determine the cartesian velocity of, for example, the right ankle as a function of the joint variables. Then we can determine the Jacobian by building the partial derivatives.

  $$v = J(\theta)\dot{\theta} \tag{2.23}$$

  If J(X) is not singular and quadratic, we can invert the Jacobian to calculate joint rates from given velocities:

  $$\dot{\theta} = J^{-1}(\theta)v \tag{2.24}$$

  Locations where the Jacobian becomes singular are called *singularities of the mechanism*. The Jacobians are used to make the transition between *work* and *configuration space* of the robot.

3 Trajectory generation (Planing)

In this section the generation of a walking pattern for a biped is going to be discussed. As a biped robot tends to tip over easily the most important constraint on the walking pattern is stability. Another constraint is given by the actuators of the robot that cannot produce unlimited torques and joint velocities. Therefor another important consideration is how we can achieve a walking motion that minimizes a cost function of energy consumption.
Several methods have been proposed for planing an optimized walking pattern for static and dynamic stability. Some specify low energy reference trajectories, while others propose a walking pattern syntheses based on the ZMP. In order for a biped robot to perform stable dynamic motions, its ZMP has to be in the convex hull of all contact points between its feet and the ground. This convex hull is also called the stable region. One method proposed consists in designing first a desired ZMP trajectory and to generate the hip and torso motion that achieves this ideal trajectory. The problem consists in the fact that not all ZMP trajectories can be achieved because the hip acceleration and the energy consumption caused by the relative massive torso might be too high. Therefor it is better to obtain the hip motion without first designing an ideal ZMP path.
The biped must be capable of various foot motion. It has to be able to lift its feet to negotiate obstacles in its way or to support the feet with a suitable angle to match the roughness of the terrain. These various parameters pose another set of constraints for gait generation and must be taken into consideration, especially if the robot will have to deal with uneven terrain. Most methods propose a trajectory generation by polynomial interpolation. As the number of constraints gets very high (ground condition, foot motion) the polynomial computed might cause oscillation of the trajectory.
A method of walking pattern generation proposed by [Huang, 2001] that focuses on dynamic stability constraints is going to be presented here, it uses spline interpolation to generate the trajectory and allows to define a set of parameters that can be adjusted according to the terrain conditions. It begins by stating the constraints on the foot trajectory. Then it generates a smooth hip motion that guarantees the largest possible stability margin. It derives the trajectory by iterative computation without first designing an ideal ZMP trajectory.

- Walking Cycle
  Biped walking is a periodic phenomena. A walking cycle is composed of two phases: a *double-support* and a *single support phase.* While both feet are on the ground during the double-support phase, only one foot is stationary on the ground in the single-support phase while the other foot swings from the rear to the front. During the short double-support phase (about 20% of the walking cycle) the robot's CoM in the static case or its ZMP in the dynamic case must be transferred from the rear foot to the front foot.
  The walking cycle and the according angle-position of the feet is shown in the figure below. If both foot and hip trajectories are known, all joint trajectories of the biped robot can be obtained by kinematic constraints. The walking pattern can therefore be denoted uniquely by the trajectories of the feet and the hip. As the lateral hip motion can be obtained similarly to the sagittal hip motion, we will only discuss trajectories in the sagittal plane here.

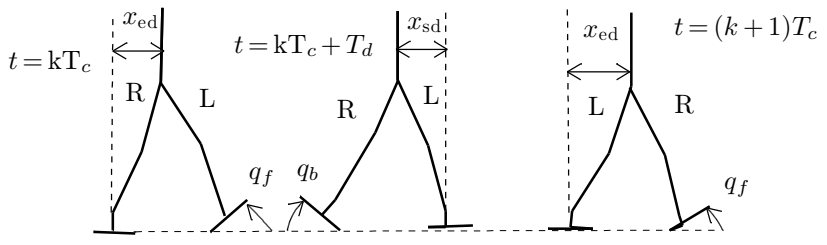

figure3.1

(source: 11)

- Foot Trajectories
  We define the step cycle for the right foot according to the figure above. The left foot trajectory can be obtained accordingly. We define the k-th walking step to begin with the heel of the right foot leaving the ground at $t = \mathrm{kT}_c$ and to end with the heel of the same foot making first contact with the ground at $t = (k+1)T_c$. In the following we will assume the ground to be flat for simplicity. By adding some additional parameters the same analysis can be done for an uneven surface.
  The following parameters are necessary for formulating the constraints of the foot trajectory:

T_c	step length
T_d	length of double support phase
T_m	time when right foot is at the highest point
D_s	step width
q_b, q_f	angles of foot leaving, landing on the ground (see figure above)
H_a, L_a	coordinates of the highest point (see figure below)
$l_{\mathrm{an}}, l_{\mathrm{af}}, l_{\mathrm{ab}}$	foot geometry (see figure below)

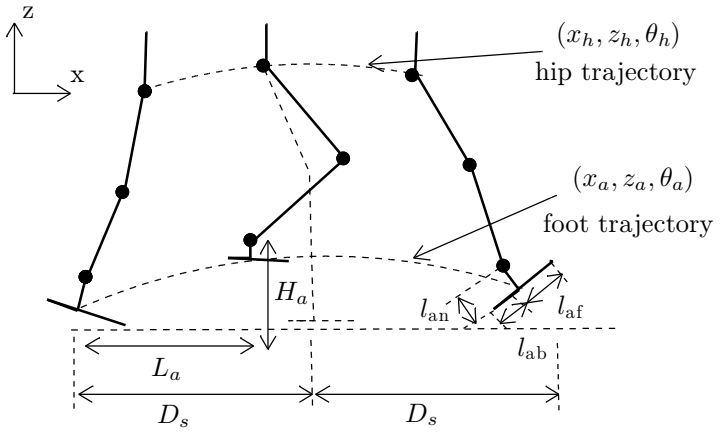

figure3.2

(source: 11)

Now we can formulate the constraints on x_a, z_a and θ_a. To generate a smooth trajectory, it is necessary that the first derivative (velocity) terms be differential and the second derivate (acceleration) terms be continuous at all t, including the breakpoints given in the equations below. We obtain the trajectories by third-order spline interpolation. In this case, x_a, z_a and θ_a are characterized by third-order polynomial expressions, and the second derivatives are always continuous. By varying the parameters H_a, L_a, q_b and q_f we can produce different foot trajectories.

(3.1)

breakpoints↓ parameters →	$\theta_a(t)$	$x_a(t)$
$t = \mathrm{kT}_c$	0	kD_s
$t = \mathrm{kT}_c + T_d$	q_b	$\mathrm{kD}_s + l_{\mathrm{an}}\mathrm{sinq}_b + l_{\mathrm{af}}(1 - \mathrm{cosq}_b)$
$t = \mathrm{kT}_c + T_m$	-	$\mathrm{kD}_s + L_a$
$t = (k+1)T_c$	$-q_f$	$(k+2)D_s - l_{\mathrm{an}}\mathrm{sinq}_f - l_{\mathrm{ab}}(1 - \mathrm{cosq}_f)$
$t = (k+1)T_c + T_d$	0	$(k+2)D_s$

breakpoints↓ parameters →	$z_a(t)$	$\dot{\theta_a}(t)$	$\dot{x_a}(t)$	$\dot{z_a}(t)$
$t = \mathrm{kT}_c$	l_{an}	0	0	0
$t = \mathrm{kT}_c + T_d$	$l_{\mathrm{af}}\mathrm{sinq}_b + l_{\mathrm{an}}\mathrm{cosq}_b$	-	-	-
$t = \mathrm{kT}_c + T_m$	H_a	-	-	-
$t = (k+1)T_c$	$l_{\mathrm{ab}}\mathrm{sinq}_f + l_{\mathrm{an}}\mathrm{cosq}_f$	-	-	-
$t = (k+1)T_c + T_d$	l_{an}	0	0	0

- Hip Trajectory
  From a viewpoint of stability it is desirable that hip motion parameter $\theta_h(t)$ is constant, in particular $\theta_h = 0.5\pi$ rad on level ground. Hip motion $z_h(t)$ hardly affects the position of the ZMP. We can specify $z_h(t)$ to be constant, or to vary within a fixed range. $x_h(t)$ is the main factor that affects the stability of the biped robot walking in a sagittal plane (as we said before, for reasons of simplicity we do not consider $y_h(t)$, the motion in the lateral plane). As we do not want to calculate a desired ZMP trajectory, we will proceed by first calculating a series of smooth x_h-trajectories and then determine $x_h(t)$ that guarantees the largest stability margin.
  The figure below illustrates the stability margin. If the minimum distance between the ZMP and the boundary of the stable region is large, the moment preventing the robot from tipping over is large. d_{zmp} is the minimum distance between the boundary of the stable region and the ZMP. It is called the stability margin. The equation for calculating the ZMP will be presented later.

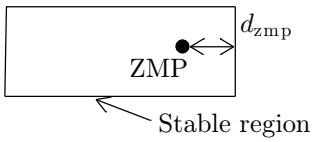

figure3.3

(source:11)

During a one-step cycle, $x_h(t)$ can be described by two functions for the double and single-support phase respectively. The hip trajectory will be a function of only 2 parameters, that we define as x_{sd} and x_{ed} (see figure 3.1). They denote distances along the x-axis from the hip to the ankle of the support foot at the start (x_{sd}) and end (x_{ed}) of the single-support phase. We get the following constraints on the $x_h(t)$:

(3.2)

breakpoints↓ parameters →	$x_h(t)$
$t = \text{kT}_c$	$\text{kD}_s + x_{\text{ed}}$
$t = \text{kT}_c + T_d$	$(k+1)D_s - x_{\text{sd}}$
$t = (k+1)T_c$	$(k+1)D_s + x_{\text{ed}}$

To guarantee a smooth periodic $x_h(t)$, the following derivative constraints must be satisfied:

$$(3.3) \qquad \dot{x_h}(\text{kT}_c) = \dot{x_h}(\text{kT}_c + T_c) \quad \wedge \quad \ddot{x_h}(\text{kT}_c) = \ddot{x_h}(\text{kT}_c + T_c)$$

We use 3rd-order periodic spline interpolation to obtain a trajectory that satisfies the constraints (3.2) and (3.3). We calculate a series of smooth hip trajectories by varying the two parameters x_{sd} and x_{ed} and chose $x_h(t)$ with the largest stability margin d_{zmp}.

$$(3.4) \qquad \max\,[d_{\text{zmp}}(x_{\text{sd}}, x_{\text{ed}})], \quad x_{\text{ed}} \,\epsilon\, (0, 0.5D_s), x_{\text{sd}} \,\epsilon\, (0, 0.5D_s)$$

Since there are only two parameters x_{ed} and x_{sd}, solutions for (3.4) can be easily obtained by exhaustive search computation. We can now calculate the corresponding trajectories of the joint angles and velocity rates using inverse kinematics as explained in the previous section for the corresponding time interval.
In the next section we will compute the equations of motion describing the biped mechanism. This will enable us to determine the torques of the actuators necessary to realize the planned trajectories.

4 Equations of Motion (Actuating)

In this section we will derive the equations of motion of the biped robot. The robot is a MBS, so we can compute the EOM recursively by using the *Newton-Euler Method* or the *Lagrangian dynamic formulation.* Whereas the Newton-Euler formulation is a "force balance" approach to dynamics, the Lagrangian formulation might be considered as an "energy based" approach. Both methods will yield the same equations of motion.
Computing the equations of motion gives us the possibility of predicting the dynamic behavior of our system. In the previous section we described a reference trajectory for the configuration space of the robot by using inverse kinematics to map the desired walking motion of the biped robot on the joint angles and angular rates. By means of the EOM, we will now be able to state whether the desired trajectory can be realized by the technical restrictions of the robot or not. Using the EOM, we compute the necessary joint torques to realize the walking motion. Then we analyze the obtained torque results. If the required torques are within the performance limits of the actuators (joint motors) and if several other constraints (as angular velocity limitations, ZMP or friction conditions) are fulfilled, we have obtained a trajectory pattern that can be realized by the biped robot.

This chapter is surnamed *Actuating*, because it yields the joint torques necessary for the desired biped locomotion. If there were no external disturbances and if our model of the robot would be perfect, the work would be done already and we could relax. But because our modeling is only an imperfect imitation of the real object and as there are always a lot of parameters that we have not included in our model, we will have to consider how we can guarantee the biped's stability while he is actually walking. *Controlling* considerations will be discussed in the next chapter.

- Equations of motion
  As stated before, we use either recursively the *Newton-Euler* method or the *Lagrangian dynamic formulation* in order to obtain the equations of motion of the MBS that constitutes our model of a biped robot. If we express the dynamics of the robot in a single equation and evaluate the *Newton-Euler* equations symbolically, they yield a dynamic equation that can be written in the form

$$M(\theta)\theta^{\cdot\cdot} + V(\theta,\theta^{\cdot}) + G(\theta) = Q_{\text{mot}} + Q_{\text{feet}} + Q_{\text{friction}} \tag{4.1}$$

The left hand side of (4.1) incorporates all conservative forces, while the right hand side are the generalized forces resulting from motor torques, ground contact forces and gear friction. The vector θ (joint angles) represents the generalized coordinates. Defining θ as the generalized coordinates implies neglecting the torques of the joint motor armatures. Still, considering the fact that the harmonic drive gears used for actuating the joints of a biped robot have very high gear ratios, the motor armature torques are very small compared to the torques produced at the robot joints. This is why, for simplicity reasons, we will not consider them here.
$M(\theta)$ is the n x n *mass matrix* of the manipulator (n being the number of generalized coordinates that describe the MBS). $M(\theta)$ must be symmetrical. $V(\theta, \theta^{\cdot})$ is a n x 1 vector of centrifugal and Coriolis terms and $G(\theta)$ is an n x 1 vector of gravity terms. We use the term *state-space equation* to describe this form of the EOM because the $V(\theta, \theta^{\cdot})$ vector has both position and velocity dependence.
By writing the velocity-dependant term $V(\theta,\theta^{\cdot})$ in a different form, we can restate the dynamic equations as

$$M(\theta)\theta^{\cdot\cdot} + B(\theta)[\theta^{\cdot}\theta^{\cdot}] + C(\theta)[\theta^{\cdot 2}] + G(\theta) = Q_{\text{mot}} + Q_{\text{feet}} + Q_{\text{friction}} \tag{4.2}$$

$B(\theta)$ is a n x n(n-1)/2 matrix of Coriolis coefficients and $C(\theta)$ is a n x n matrix of centrifugal

components. $[\theta^{\cdot}\theta^{\cdot}]$ and $[\theta^{\cdot 2}]$ have the forms $[\theta^{\cdot}{}_1\theta^{\cdot}{}_2 \quad \theta^{\cdot}{}_1\theta^{\cdot}{}_3 \quad .... \quad \theta^{\cdot}{}_{n-1}\theta^{\cdot}{}_n]\top$ and $[\theta^{\cdot 2}_1 \; \theta^{\cdot 2}_2 \; .... \; \theta^{\cdot 2}_n]\top$ respectively.
This way of writing the dynamics equation is called the *configuration-space equation*, because matrices are functions only of the position of the robot. As all parameters in this form of the equation are only dependant on θ, the computing of the EOM becomes easier. This is important, because if we want to control the biped robot later on, we will have to update the EOM as the biped walks and thus depend on a reliable and fast computation.

- Generalized forces
As mentioned before, the generalized forces in the EOM result from motor torques, ground forces and gear friction. A short explanation is given how we can obtain the generalized forces for motor torques and gear friction. Because we want to compute the torques necessary for the realization of the planned walking motion, it is very important to properly model the contact forces resulting from the contact of the robot's feet with the ground. A way of simulating the foot-ground contact and of computing the resulting contact forces will be shown thereafter.

  → Motor generalized forces (Q_{mot})
  The motor dynamics can be described as

  $$T_{\text{mot}_i} = k_{\text{mot}_i} I_i \tag{4.3}$$

  with k_{mot} being the torque constant and I_i the motor current. Given the gear ratio η of the joint, we can compute the corresponding joint torque

  $$\tau_{\text{mot}_i} = \eta T_{\text{mot}_i} \tag{4.4}$$

  The generalized forces resulting from the applied armature voltage are

  $$Q_{\text{mot}} = \sum_{i=1}^{n_{\text{drive}}} \left(\frac{\partial \theta^{\cdot}{}_i}{\partial \theta^{\cdot}}\right)\top \tau_{\text{mot}_i} = \sum_{i=1}^{n_{\text{drive}}} \tau_{\text{mot}_i} \tag{4.5}$$

  → Gear (friction) generalized forces(Q_{friction})
  Nonrigid body effects are important to include in the model, because forces due to friction can actually be very large (up to 25% of the torque required in some situations). Here, a simple example of friction inclusion is presented. It shall comprise viscous friction ($\tau_{\text{friction}} = k\theta^{\cdot}$, $k :=$ viscous friction constant), Coulomb friction ($\tau_{\text{friction}} = \text{csign}(\theta^{\cdot})$, $c :=$ static/dynamic coefficient) and Stribeck friction. Instead of supplying a formula, the resulting friction curve is approximately shown below in dependency of the joint angular velocity:

  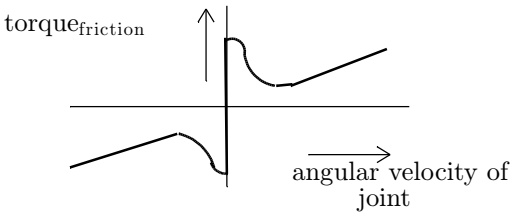

  figure4.1

  Now we can formulate the generalized forces resulting from the non-rigid body effects:

  $$Q_{\text{friction}} = \sum_{i=1}^{n_{\text{drive}}} \left(\frac{\partial \theta^{\cdot}{}_i}{\partial \theta^{\cdot}}\right)\top \tau_{\text{friction}_i} = \sum_{i=1}^{n_{\text{drive}}} \tau_{\text{friction}_i} \tag{4.6}$$

  → Ground contact forces (Q_{feet})
  The contact model presented here was used to calculate the contact ground forces for the biped robot JOHNNIE. It models the impact damping mechanism that takes place when the robot sets its right or left foot on the ground. In the case of JOHNNIE, there are 4 contact elements at each foot that consist of a high friction rubber layer and damping material. The contact element model is shown below:

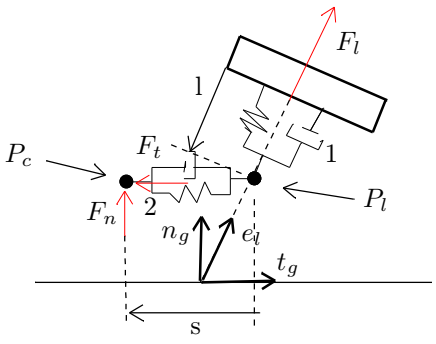

figure4.2

(source: 3)

The damper and spring coefficients of (1) shall be K_l and B_l, those of system (2) K_s and B_s. As the figure suggests, the foam damping material and the vertical compliance of the rubber layer are modeled as a linear spring and damper system. The ground contact is simulated as unilateral point contact ($F_n \geq 0$) with Coulomb friction. Here only the force calculated in the case of stiction is going to be mentioned. It has been established in experiments that the foot of the biped robot does not slip, at least when the robot is walking at a regular speed of about 2km/h. Still, when we simulate the ground contact forces, we first compute F_n assuming stiction. Then if the equation

(4.7) $\text{abs}(F_t) \leq \mu F_n$

is not satisfied, we recompute F_n for slippage by applying the law of coulomb friction. It is implicitly assumed that (4.7) is satisfied here.
The equilibrium of forces in e_l direction yields:

(4.8) $F_l = \boldsymbol{F}_n \top \boldsymbol{e}_l + \boldsymbol{F}_t \top \boldsymbol{e}_l$

F_t and F_l are given by:

(4.9) $\boldsymbol{F}_t = -K_s \boldsymbol{s} - B_s \dot{\boldsymbol{s}}$ (4.10) $\boldsymbol{F}_l = -K_l \boldsymbol{l} - B_l \dot{\boldsymbol{l}}$

Now we solve for F_n. The solutions yields (we use $\boldsymbol{F}_n = F_n \boldsymbol{n}_g$):

(4.11) $F_n = \frac{-K_l l - B_l \dot{l} + \boldsymbol{e}_l \top (K_s \boldsymbol{s} + B_s \dot{\boldsymbol{s}})}{\boldsymbol{n}_g \top \boldsymbol{e}_l}$

The kinematic parameters l and $\dot{l}$ are given by the trajectory of the feet that we have planned in advance. s and $\dot{s}$ are evaluated from the relative displacement between P_l and P_c. The total contact force F_c for every foot is given by (4.12):

(4.12) $F_c = \sum_{i=1}^{4} F_n$

(4.12) is valid because, in the case of JOHNNIE, we got 4 contact elements per foot. (4.13) yields the resulting generalized forces:

(4.13) $Q_{\text{foot}} = \sum_{i=1}^{2} (\frac{\boldsymbol{v}_{\text{foot}_i}}{\partial \dot{\boldsymbol{\theta}}}) \top \boldsymbol{F}_{c_i}$

Q_{foot} only has to be taken into account during double support phase. During single support, the contact forces of the supporting foot do not have to be considered, because its velocity-vector equals zero.

- Dynamic constraints

  In order to realize a stable dynamic motion, the following constraints must be satisfied by the EOM:

  → Kinematic constraints
  During swing-phase, the mechanism of the biped forms an open kinematic chain. During pre-swing and heel-contact, however, both feet have ground contact, thus form a closed kinematic chain. This is taken into account by kinematic constraints imposed on the system acceleration. We use the following coordinate system: x points along the axis in the sagittal plane, y corresponds to the axis in the lateral plane. The constraints of the pre-swing phase for the not supporting foot are:

  $$(4.14)\quad \boldsymbol{v}_{\text{toe}} = \begin{pmatrix} {r_f}^{\cdot\cdot}(t) \\ {\omega_x}^{\cdot\cdot}(t) \\ {\omega_z}^{\cdot\cdot}(t) \end{pmatrix} = J_f(\theta(t))\theta^{\cdot\cdot}(t) + J_f{}^{\cdot}(\theta(t))\theta^{\cdot}(t) = 0$$

  $\boldsymbol{v}_{\text{toe}}$ is a 3 x 1 vector consisting of angular and radial acceleration. The equation implies that there are no roll and yaw motions of the rear foot during the pre-swing phase. (4.14) is the derivative form of the Jacobian mapping of joint rates to velocities (2.23). The Jacobian must be computed relative to the foot frame. The same constraint applies for the heel-contact-phase when the swinging foot touches the ground. Here the acceleration of the heel must be zero. (4.15) formulates the constraint.

  $$(4.15)\quad \boldsymbol{v}_{\text{heel}} = \begin{pmatrix} {r_f}^{\cdot\cdot}(t) \\ {\omega_x}^{\cdot\cdot}(t) \\ {\omega_z}^{\cdot\cdot}(t) \end{pmatrix} = J_h(\theta(t))\theta^{\cdot\cdot}(t) + J_h{}^{\cdot}(\theta(t))\theta^{\cdot}(t) = 0$$

  → Zero Moment Point constraints
  The ZMP can be computed using (4.16) and (4.17):

  $$(4.16)\quad x_{\text{zmp}} = \frac{\sum_{i=1}^{n} m_i({z_i}^{\cdot\cdot} + g)x_i - \sum_{i=1}^{n} m_i {x_i}^{\cdot} z_i - \sum_{i=1}^{n} \text{Iiy}{\theta_{\text{iy}}}^{\cdot\cdot}}{\sum_{i=1}^{n} m_i({z_i}^{\cdot\cdot} + g)}$$

  $$(4.17)\quad y_{\text{zmp}} = \frac{\sum_{i=1}^{n} m_i({z_i}^{\cdot\cdot} + g)y_i - \sum_{i=1}^{n} m_i {y_i}^{\cdot} z_i - \sum_{i=1}^{n} \text{Iix}{\theta_{\text{ix}}}^{\cdot\cdot}}{\sum_{i=1}^{n} m_i({z_i}^{\cdot\cdot} + g)}$$

  Using an inverted pendulum model for the robot, we will be able to calculate the ZMP with a less complicated formula. This will be shown in the next section.
  The following figure illustrates the feasible regions for the ZMP (top view):

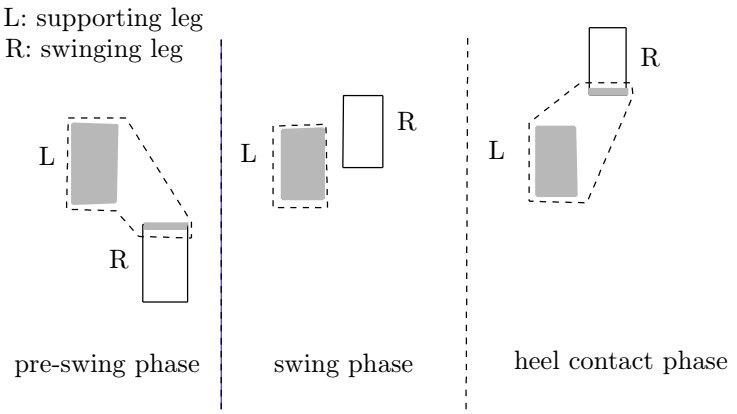

figure4.3

(source: 8)

5 Control

In this chapter, controlling issues will be discussed and illustrated at the example of the biped robot JOHNNIE. From the viewpoint of control and walking pattern generation, the research on biped humanoid robots can be classified into two categories. The first group requires the precise knowledge of robot dynamics including mass, location of CoM and inertia of each link to prepare the walking patterns. The method proposed is called *Computed Torque Control* and relies on a non-linear feedback system.
The other category might be called inverted pendulum approach, since the inverted pendulum model is used. This method uses only limited knowledge of dynamics, e.g. location of total CoM or total angular momentum. The method proposed is called *joint position control*.
Both approaches have been implemented on JOHNNIE. In order to understand the concepts used for the dynamic control, a short illustration of the *Inverted Pendulum Model* will be given first. Based on this model, we will see how to control the ankle joint torque of the biped walking machine in order to achieve a stabilized walking motion.

- Inverted pendulum model
  In this approach, we make the assumption that the stability behavior of our walking biped can be compared to an inverted pendulum. In order to do so, we make use of a lumped mass model of the robot. Thus, the robot appears as a pendulum with its mass lumped in its CoM. For simplicity, we can assume that the CoM of the robot is located at the pelvis.
  We apply a constraint control on the inverted pendulum such that the mass should move along a horizontal plane with a height that corresponds to the z-coordinate of the pelvis z_p.
  Now we compute the torque balance (see figure below):

(5.1) $\mathrm{mz}_p\ddot{x} = \tau_y + \mathrm{mgx}$ -> (5.2) $\ddot{x} = \frac{\mathrm{gx}}{z_p} + \frac{\tau_y}{\mathrm{mz}_p}$

(5.2) $\mathrm{mz}_p\ddot{y} = -\tau_x + \mathrm{mgy}$ -> (5.4) $\ddot{y} = -\frac{-\tau_x}{\mathrm{mz}_p} + \frac{\mathrm{gy}}{z_p}$

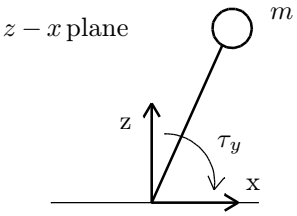

figure5.1

(source: 7)

As the contact forces on the ground result from the robot's inertia and gravitational forces, we can compute the position of the ZMP δ only by considering the ground reaction forces.

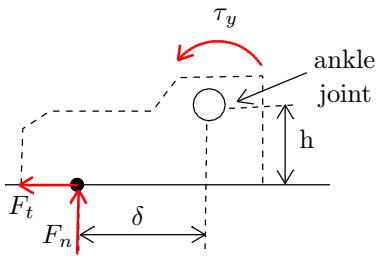

figure5.2

(source: 7)

(5.5) $\tau_y = F_n\delta + F_t h$

or (neglecting ankle height h)

(5.6) $\tau_y = F_n\delta = \mathrm{mg}\delta$

Now we can formulate δ in dependency of the torque acting on the robot's leg:

(5.7) $\delta_x = -\frac{\tau_y}{\mathrm{mg}}$ (5.8) $\delta_y = \frac{\tau_x}{\mathrm{mg}}$

We replace (5.7/8) in equations (5.2/4) and finally obtain a relationship that shows the dependency between CoM acceleration and ZMP:

(5.9) $y^{\cdot\cdot} = \frac{g}{z_p}(y - \delta_y)$ (5.10) $x^{\cdot\cdot} = \frac{g}{z_c}(x - p_x)$

(5.11) $\delta_y = y - \frac{z_p}{g}y^{\cdot\cdot}$ (5.12) $\delta_x = x - \frac{z_p}{g}x^{\cdot\cdot}$

The inverted pendulum serves as a good example for dynamic walking. Even if the projection of the CoM is not inside the contact polygonal, stable walking is possible because, by varying the acceleration of the CoM, the ZMP can be controlled so as to remain inside the stable region. The following figure illustrates ZMP and CoM trajectory during the swing-phase of a biped robot. The acceleration changes when the swinging foot sets on the ground.

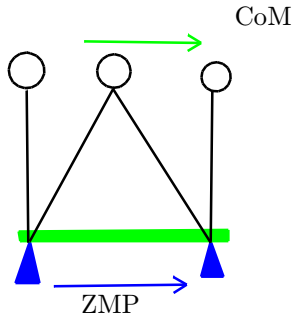

figure5.3

(source: 9)

- Ankle joint torque control
  In order to control the overall posture of the robot an online adaption of the trajectories is necessary. The control system used on the biped robot JOHNNIE was a *Feedback-Linearization* system. Later, it was replaced by a linear *Joint Position Control* system. Basing on the trajectory computation, the velocities are mapped into joint space using inverse kinematics. The reference joint angular rates and velocities are fed to the micro-controllers that act as PID-controllers. Here, only the control mechanism of the ankle joints is presented.
  The robot JOHNNIE is equipped with 2 six-axes force/torque sensors. These sensors consist of a single aluminum part. 3 deformation beams within the load path hold strain gauges. The maximum torque that can be transmitted is limited by the foot geometry and causes a small margin of stability.

  (5.13) $\mathrm{abs}(\tau_x) \leq 0.5F_n l_y$ (5.14) $\mathrm{abs}(\tau_y) \leq 0.5F_n l_x$

  l_x and l_y describe the foot geometry.
  As we have seen before, we can compute the actual ZMP if we know the acceleration of the CoM. In order to measure the position and acceleration of the robot pelvis, JOHNNIE is equipped with an attitude sensor that consists of an accelerometer and gyroscopes. Its bandwidth is about 85Hz. We will use the attitude sensor to compute the deviation of the ZMP from its ideal trajectory(see figures 4.3 and 5.3).
  In order to stabilize the robot, the ankle joint is used to control the foot torques. As we have

seen in figure 5.2, we can control the position of the ZMP via the torque acting on the ankle joint. If we measure a deviation of the ZMP from the projected trajectory, we prevent the robot from tipping over by varying the torque on the ankle joint so that the robot can keep an upright posture.
In order to model the compliance of the ground, the stiffness of the links and the elasticity of the foot elements we use a contact stiffness C_F. Now the torque transmitted from the foot on the ground can be formulated as

$$\tau_F = -C_F(\varphi_o + \varphi_F) \tag{5.15}$$

φ_F is illustrated in the figure below (φ_o is the inclination of the upper body):

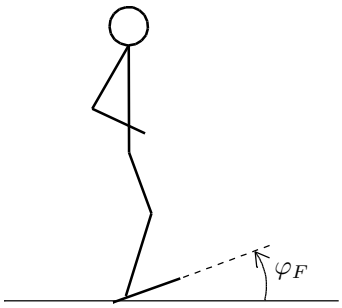

figure5.4

(source: 4)

Thus, by actuating the pitch and roll motion of the feet, we can act on the ankle torque and thereby stabilize the robot. In the case of JOHNNIE, the ankle joint is actuated by two parallel ballscrew drives.
One problem is the limited bandwidth of the attitude sensor. This means that we can not depend uniquely on the signals of the attitude sensor, because the signal flow would be to slow in order to achieve a stable and robust control. In order to solve this problem, we implement a servo-loop inside the outer-loop. Hence, we have a fast inner servo-loop, consisting simply of multiplying errors by gains, with the deviation torques added at a slower rate.
The control scheme for the robot JOHNNIE is shown in the following figure. The faster inner control-loop is marked red, while the slower outer-loop is shown by a green line:

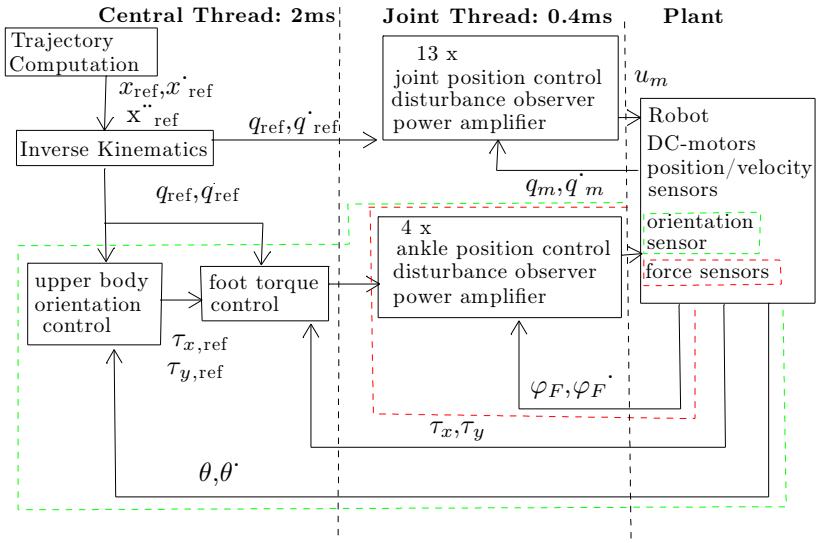

figure5.5

(source: 4)

6 Conclusion

In the previous sections, we did analyze the procedure that leads to simulating and modeling a walking biped machine.
First, we did plan a gait pattern to enable the robot to achieve a stable walking motion. We used a method that guarantees a maximum stability margin and enables us to change the gait parameters to negotiate uneven surfaces or ground irregularities. A next step would consist in computing an algorithm that enables the robot to perform autonomously a varying walking motion, e.g. the robot should decide by itself whether to step over an obstacle, or whether it is better to walk around it. This kind of autonomous trajectory pattern generation has been implemented on the biped robot JOHNNIE and was tested successfully using a visual system guidance.
In the next step, we computed the EOM that enabled us to predict the dynamic behavior of the biped mechanism.
Finally, we analyzed a way of controlling the stability of the system by using the inverted pendulum and a lumped mass model of the robot.

7 Table of contents

(Some) References:

1. John J. Craig, “Introduction to robotics”, Prentice Pearson Hall, 2005
2. F. Pfeiffer, “Entwurf und Realisierung einer zweibeinigen Laufmaschine”
3. T.Buschmann, S.Lohmeier, H.Ulbrich, F.Pfeiffer, “Dynamics Simulation for a Biped Robot”
4. S.Lohmeier, K.Löffler, M.Gienger, H.Ulbrich, ”Computer System and Control of Biped Johnnie”
5. K.Löffler, M.Gienger, F.Pfeiffer, “Sensor and Control Design of a Dynamically Stable Biped Robot”
6. M.Ahanikamangar, “Zur Kinematik und Dynamik einer zweibeinigen Laufmaschine mit Laufmustergenerator”, VDI-Fortschrittsberichte, 1991, Nr.99
7. A.Masuyama, K.Mitobe, Y.Nasu, “Ankle joint control for walking robots”
8. J.Denk, G.Schmidt, “Synthesis of a Walking Primitve Database for a Humanoid Robot using Optimal Control Techniques”, Proceedings IEEE-RAS International Conference (HUMANOIDS2001)
9. S.Kajita, F.Kanehiro, K.Kaneko, K.Fujiwara, ”Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point”, Proceedings of the 2003 IEEE, Taiwan
10. K.Hirai, M.Hirose, Y.Haikawa, T.Takenaka, “The Development of Honda Humanoid Robot”, Proceeding of the 1998 IEEE, Belgium
11. Q.Huang, K.Yokoi, S.Kajita, K.Kaneko,”Planning Walking Patterns for a Biped Robot”, IEEE, 2001