Final Year Project Report
Development and implementation dynamic balance algorithms for bipedal robot locomotion.

Usvyatsov Mikhail

Innopolis University
Kazan

Supervisor: Prof. Evgeni Magid

May 31, 2015

Contents

Chapter 1

Introduction and Overview

Nowadays humanity has invented almost all devices that are needed for modern humans and society in general. Science is now engaged in the improvement and optimization of existing solutions. These solutions can be traced with trends that are repeated and replicated. This approach is quite good: gain experience, accumulate knowledge and apply them to the latest developments. We can find these patterns in such old area as automobile industry for example. In robotics there are no such patterns yet. It means that inventions in robotics can enrich creator of these inventions. That is one of the reasons to work in robotics sphere.

The word robot was initially attributed in 1921 to fictional humanoid by Karel Capaek. Also the first use of the word "robot" refers to a humanoid machine that was supposed to serve a human. It was a common belief before the advent of industrial robots that robots should look like humans. However, from the very beginning almost every automatic device intended for production and other operations normally performed by a human was called robot. Intensive development of robots started after the Second World War, which was associated with the emergence of the nuclear industry.

The science studying robots is called robotics. Robotics considers hardware agents that are called robots. Nowadays almost every professor teaching robotics can give his own definition of robots and all of them will not be similar. One of robot today's definition is autonomous device that work in automatic mode. It can be attributed to the machine "Lunokhod-1", created in 1966. It was the first machine in the history that worked on the surface of the Moon (1970). Another modern definition is a mechanical or virtual artificial agent, usually an electro-mechanical machine that is guided by a computer program or electronic circuitry. This agents operates in the environment that is called workspace. According to [1] it is very important to consider the level of uncertainty of the environment where a robot works, because robots will interact with this environment. We can distinguish robots by their reachable workspace. From this point of view we can distinguish mobile robots whose reachable workspace is not constant and also is usually partially observable. On the other hand there are static robots, that can interact with environment only with end-effectors. They have fixed and usually fully observable workspace.

Constantly emerging information about the achievements of the leading countries in the development of military land, underwater robots and unmanned aerial vehicles supports for this thesis. Despite of high price robotic products become a part of our usual life, e.g. automatic vacuum cleaners, robotic toys, smart houses, etc. There is a tendency that robots become cheaper with constantly increasing abilities. It means that in not very far future very smart robots will become affordable for our usual life. Lately robots began to replace humans not only in manufacturing but also in the military sphere. Such robotic tools will replace humans in difficult and dangerous environments. This proposes the task to develop universal robots that can solve wide spectrum of tasks in the same environment as human works in. If we want a mechanism to perform its actions in environment that was developed for people it is an obvious solution to make this mechanism similar to human. Robots that have similar to human kinematics and appearance are called anthropomorphic robots. Examples of such robots are well known. They appear in films, literature and now they become part of our life, e.g. professor Ishiguro developed a robot that is similar to him and during his trips this robot conducts professor's lectures while he can be on another part of Earth. Anthropomorphic robots now can be divided into two big group. Bipedal anthropomorphic robots and other anthropomorphic robots. Bipedal robots are the robots that perform their locomotion by two legs. It is important to contrast bipedal anthropomorphic robotics because of many reasons. One of them is that bipedal humanoid walking robots comprise the area of robotics that is developing most rapidly nowadays. It is reasoned by the following fact: human workspace includes very complex and uneven terrains which are hard to overcome all the other ways except by legs. On the other hand bipedalism allows creatures to inhabit and adopt to very wide range of environments. Our initial wish was to design a robot able to work in the same workspace with people. On the other hand, in the last century robotics is mainly focused on space and defense areas. It motivates the trend towards automation as a core part of progress. Automation reduces the cost of technical processes and the risk to humans. Therefore, the research and development for this task are on the cutting edge of science and technology and require special attention and investment into its development. There are more and more situations requiring people to perform a wide variety of work in heavy, dangerous, and sometimes incompatible with life conditions. In response, there are new tools of extreme robotics. However, for the most part they are very similar to each other. Usually, autonomous wheeled or tracked vehicles with manipulators are used to perform tasks on the ground. Mostly these robots are teleoperated. These robots are being produced for more than a dozen years. Engineers have so far accumulated a lot of experience in their development and applications, which are in some cases very efficient. However, it is obvious that this technique has (like any other) a limited scope of application. People still have to work in dangerous conditions such as in chemical, biological, or radioactive hazards, work in extremely hot or cold conditions, fight against criminals and terrorists. Human workspace is very specific due to people's two arms and two legs. A universal robot should operate in the same environment and workspace. For this reason other areas of robotics are developing now.

One of these is a robotic system including an anthropomorphic bipedal walking robot. Such robot's kinematics, size and weight are similar to human's, it is equipped with an energy source, communication channel with the control station, and a powerful autonomous control system, allowing it to perform actions in supervised or automatic mode. For instance, autonomous actions include independent movement from place to place in absence of communication. Such robots have significant advantages in workspaces initially adapted for humans.

However nowadays there are still only limited number of robots able to walk. Furthermore these robots are too expensive to let them go out of the laboratory. It is reasoned by the fact hat humanoids are extremely complex dynamical non-linear systems. The last fact means that there is no closed-form solution for control such robots. It can be explained by several reasons:

1. Humanoids are undereducated due to inertia frame. Inherently, at each contact

point, a conversion from internal joint forces to external reaction forces is needed by the humanoids interaction with its environment [2].

2. Humanoids are multybody systems with very big amount of Degrees of Freedom (DoF). Thus their dynamics require a complicated coordinate frame handling [2].

3. Humanoids structure is not constant. It consists of three various kinematic chains. Dealing with such situations in a conventional way requires the robot to plan all potential link structures beforehand and shift between them while executing [2]

On the other hand humanoids have some fundamental characteristics regardless of kinematic structure [3]. Humanoids have one one extra passive DoF that is described by ability to rotate around any of the foot boundaries. Also the gait of humanoids is usually symmetrical. Finally, biped locomotion systems interchange single- and double-support phases at a fixed interval. In the statically-stable double-support phase the bipedal robot is supported by both feet while in the statically-unstable single- support phase only one foot contacts the ground and the other swings from back to front. This explains the structure-varying principle as the humanoid alters its structure from an open to a closed kinematic chain throughout one walking cycle. [4]

Bipedal locomotion is considered by different science theories and cannot be described by only one of them. Robotics, Cybernetics, Biomechatronics, AI are only several of prospectives that a required to take into account in bipedal robots development.

Bipedal locomotion research is motivated by several reasons. They are better understanding of human locomotion, bipedal robot ability in rough terrain.

Any attempt to build artificial bipedal robot require simplified model of robot. Humans possess approximately 350 muscle pairs for walking. That involves a very complex dynamics.

The goal of this thesis is to investigate locomotion principles and to develop a control method for simulated humanoid robots. The root of this control method is based on the Zero-Moment Point, the prominent dynamic stability criterion which has been thoroughly formulated, studied and used in the last 30 years in the robotics research community [4]. Due to the high complexity, the humanoid locomotion still remains challenging and has plenty more to advance in the future.

Chapter 2

Literature review

There are several works that introduce different approaches for solving bipedal locomotion problem. This chapter will try to review the most popular and perspective previous studies. Also it will try to classify existing approaches.

Bipedal locomotion is a very complex task and according to [5] it is described by nontrivial dynamics. Up to date It doesn't have a complete general solution. However the research of the problem of bipedal locomotion has a long history. The development of the models started from the inverted pendulum model of human walking and goes to the complex approach of actuated passive walking with ZMP control.

There are 5 groups of approaches to the problem of locomotion control [6]. These are:

1. Analytical approach.

   It uses equations of physical values to generate walking trajectories. The most popular here is ZMP criterion of dynamical stability.

2. Central Pattern Generator (CPG) approach.

   When a biologically inspired creature walk it has two different mechanisms of motion control: pattern that is produced without any adaptation mechanism and control itself. CPG is an analog of pattern generator.

3. Neural Networks approach.

   This approach is an attempt to simulate the control processes of biological creatures.

4. Hidden Markov Model (HMM) approach.

   HMMs are good for any pattern recognition, hence we can use them to recognize walking patterns in bipedal locomotion gaits.

5. Rule based approach.

   The main idea behind this approach is to create rules for all possible configurations of anthropomorphic robot. The configuration here is referred as to combination of robot parameters (joints angle configuration and for example the position of ZMP or another stability criterias). Hence the configuration is a vector, that fully defines the robot in the given time frame with predefined level of accuracy.

2.1 Analytical Approach

This approach is the oldest one. Furthermore analytical approach is most studied between all the approaches for bipedal robots dynamical stability problem. It requires the knowledge of general form that locomotion should take.

For bipedal locomotion it's required to accomplish the following steps [6]:

1. Apply stability constraints.

2. Design a gait algorithm (including double support, single support and no-support phases).

3. Solve remaining Degrees of Freedom (DoF) with Inverse Kinematics (IK).

Stability constraints are derived equations that describe conditions of robot's stability. One of such constraints is the constraint on ZMP. It will be well described further in this section. The notions of double support, single support and no-support phases are described in Chapter 3. Degree of Freedom (DoF) is the number of parameters of the system that may vary independently. Inverse Kinematics (IK) refers to the inverse kinematics problem. It can be stated as follows: given the position and orientation of a end-effector find joint angles for all the system. The end-effector here can be defined as robot device that interacts with environment.

According to [7], the most natural method that represents a human body is an inverted pendulum. The Inverted pendulum model is also one of the most common examples of analytical approach [8]. It states that control of the balance can be achieved with two components: ankle mechanism (invertors/evertors) and hip load/unloading mechanisms. Invertor, evertor and ankle mechanism are the notions from biology and are described in [9]. In different positions each of the mechanism plays a different role. Thus in tandem (intermediate position) balance is achieved by invertors/evertors mechanism while direction is controlled by hip load/unloading mechanism. It violates commonly used condition: reaction force of the floor has to go thought Center of Gravity (CoG) of the robot. Modern bio-mechanical studies show that there are angular moments around (CoG). This means that reaction force of the floor goes not though CoG and hence this approach is too approximate and it is not very accurate. CoG here is the point in a body around which the resultant torque due to gravity forces vanishes.

In [10] passive dynamics architecture was considered. Passive dynamics refers to the dynamical behavior of actuators. The word passive add the requirement to work without consumption of energy from a power supply. There robot was able to move forward at not very high constant speed and the author mentioned that the robot gait was human like. Human-like gait is desirable because of its energy effectiveness [11]. Gait is a pattern of movements of human limbs during locomotion.

Furthermore, [12] summarizes principles that allows to combine passive dynamic control with powered bipedal robots. It means that during control phase robot is powered from the power supply energy, but also it uses the gravity energy in other phases in order to reduce the load to the power supply. The results show that the gait becomes energy efficient, however it implies further work on robustness and flexibility of walking.

In 1970 Miomir Vukobratovic proposed Zero Moment Point (ZMP), a theoretical model to explain bipedal robot locomotion.
Zero moment point is a concept related to dynamics and control of legged locomotion for humanoid robots. It specifies the point with respect to which dynamic reaction force at the contact of the foot with the ground does not produce any moment in the horizontal direction, i.e. the point where the total sum of horizontal inertia and gravity forces equals to zero. Miomir Vukobratovic in [3] defines ZMP (Zero Moment Point) as a point in which we can replace all the forces and moments with one single force F_a and moment M_a respectively (see fig. 2.1). Also ZMP is a basic dynamical stability constraint. According to [13] we can divide all the existing humanoid bipedal walking robots into

two big groups: ZMP-controlled ones and passive dynamic walkers.

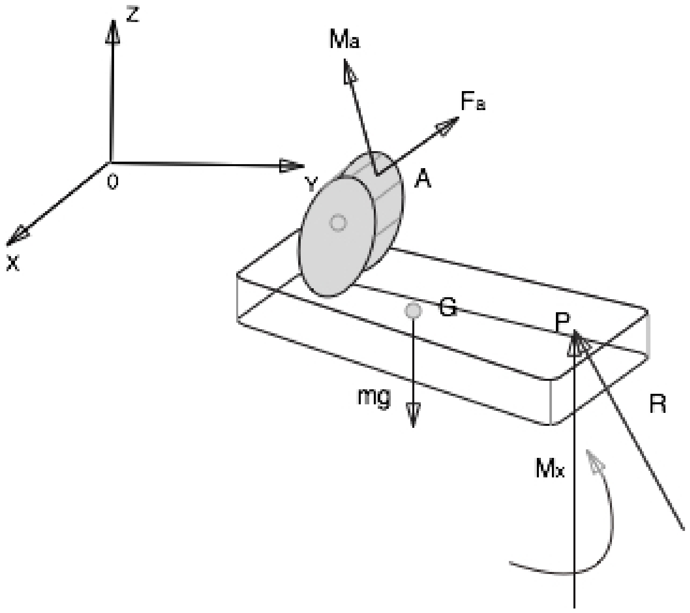

Figure 2.1: A sole with acting forces on it. mg is a gravity force. G is a center of gravity. P is a point of resulting ground reaction force application. R is a ground reaction force.

Figure 2.1 demonstrates only the sole separately from other parts of the leg. G is a Center of Gravity of the sole. At point P there is a resulting ground reaction. This ground reaction force maintains the robot in the equilibrium position. Equilibrium means that robot is not falling. The force of ground reaction R and moment M consists of its three components (R_x, R_y, R_z) and (M_x, M_y, M_z) respectively. Horizontal components of R should compensate friction force in the point of contact. Thus, the horizontal reaction of force (R_x, R_y) represents friction force that compensate horizontal component of F_a. At the same time the moment M_z represents friction reaction forces (see fig. 2.2). It compensates vertical component of M_a and the moment induced by F_a. [3]

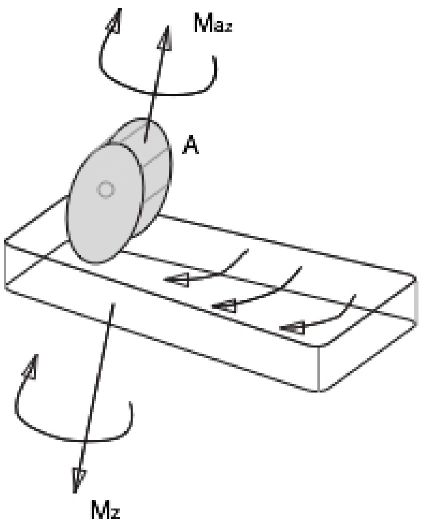

Figure 2.2: Rotational moment in the sole

ZMP should lie in the foot plane in order robot to be stable [14]. The problem is that we cannot manipulate the foot directly [15]. According to [3] we can do it by ensuring the appropriate dynamics of the mechanism above the foot. If the resulting force in ZMP lies not in vertical direction (above mentioned conditions don't hold and thus R and M_z didn't compensate correspondent components of F_a and M_a) than the foot will slide. It means that dynamical stability was not achieved due to the fact, that there is a rotational moment which affects the robot. On the other hand, in [16] it was proven that if ZMP lies in the foot area (that is usually called the support polygon) and moreover it coincides with the contact point with point P (see fig. 2.1), then the robot is stable, due to the fact, that all the resulting forces lie in the vertical direction. During the walk the position of ZMP should be computed continuously and the main problem of control is to keep ZMP and contact point to be coincided inside the support polygon of contact foot with the ground. The name zero moment point relates to the fact, that dynamical stability is maintained if horizontal components M_x and M_y are both equal to zero.

$$M_x = M_y = 0 \tag{2.1}$$

In the point P there should exist such equivalent force R and vertical moment M_z that compensate the force reaction of the ground and maintain the stability of the construction.

To achieve the dynamical stability the following equation should hold:

$$R + F_a + mg = 0 \tag{2.2}$$

Where m is a mass of the foot and g is a gravitational acceleration. O is defined In [3] as the origin coordinates frame we can define radius vectors $\vec{OP}$, $\vec{OG}$ and $\vec{OA}$ where A is a point of ankle joint.

$$\vec{OP} \times \vec{R} + \vec{OG} \times mg + M_A + M_z + \vec{OA} \times F_a = 0 \tag{2.3}$$

Placing the origin frame at point P and making a projection on the horizontal plane gives us the following equations:

$$(\vec{OP} \times \vec{R})^H + \vec{OG} \times mg + M_A^H + (\vec{OA} \times F_a)^H = 0 \tag{2.4}$$

Where H denotes horizontal component.

Equation 2.4 represents the foot equilibrium [3]. However it doesn't solve the problem, because it is still unknown whether for the given motion of mechanism it (mechanism) is kept in the equilibrium. It is only if ZMP lies inside the support polygon.

Thus, we have to make the following assumptions in order to compute the position of ZMP [17]:

1. The bipedal robot consists of n rigid links.
2. All kinematic information, such as position of Center of Mass (CoM), link orientation, velocities, etc. are known and calculated by forward kinematics.
3. The floor is rigid and motionless.
4. The feet cannot slide with regard to the floor surface.
5. All joints are actively actuated.

With this constraints we can define the mass of the robot as:

$$m_{robot} = \sum_{i=1}^{n} m_i \tag{2.5}$$

Schematic bipedal robot with math m was considered in [17] in order to derive the coordinates of ZMP (fig. 2.3).

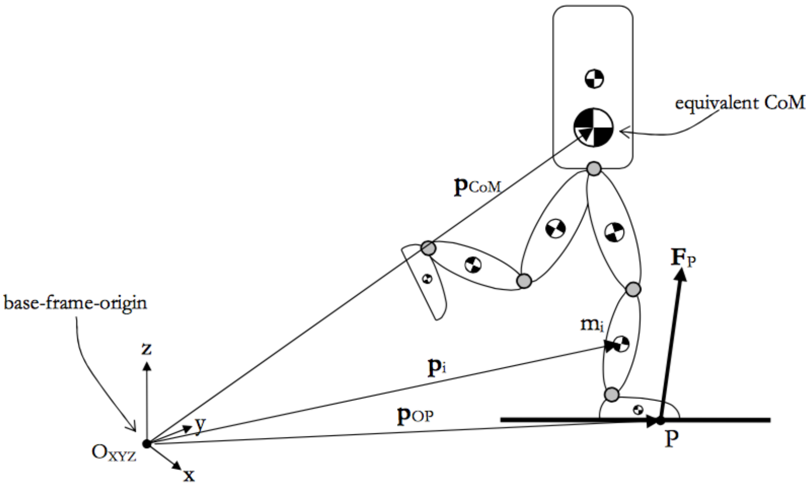

Figure 2.3: Rotational moment in the sole [17]

p_i are the distances between base frame and equivalent center of mass of i-link (see fig. 2.3). Then total linear momentum P given that $\dot{p_i}$ is p_i velocity can be described by 2.6.

$$P = \sum_{i=1}^{n} m_i \dot{p_i} \tag{2.6}$$

And total angular momentum H is:

$$H = \sum_{i=1}^{n} \dot{p_i} \times m_i \dot{p_i} + I_i \omega_i \tag{2.7}$$

where ω_i is an angular velocity and I_i is inertia tensor of link i which is computed as:

$$I_i = R_i I_i' R_i^T \tag{2.8}$$

where R_i is a rotation matrix that maps i-link coordinate frame w.r.t. the origin base frame, R_i^T is transposed R and I_i' is a inertia matrix of i-link w.r.t. the link frame origin attached to their links.

Taking the derivative of 2.6 and 2.7:

$$\dot{P} = \sum_{i=1}^{n} m_i \ddot{p_i} \tag{2.9}$$

$$\dot{H} = \sum_{i=1}^{n} \dot{p}_i \times m_i \dot{p}_i + \dot{p}_i \times m_i \ddot{p}_i + I_i \dot{\omega}_i + \omega_i \times I_i \omega_i \tag{2.10}$$

where dots represents time derivatives.

ZMP approach gives us the solution that is based on the principle of dynamical stability, however it is not energy efficient [13]. It requires simultaneous control over all the joints of the robot. The method that was described in [18] is called passive-walker dynamics and it uses gravity forces to reduce the amount of necessary energy to control the robot.

In order not to loose the balance and not to fall active control of the robot should be performed with applying dynamical stability principle. So, it makes sense to apply passive-walker dynamics with ZMP based control.

The most important task in the bipedal locomotion is to maintain dynamical stability [3]. It can be accomplished if the foot have a full contact with the ground, i.e. the contact is not only on the edge or at the point. Moreover it shows that ZMP position depends on the robot dynamics: the resulting force in the contact polygon and total moment in ZMP. So, during the motion the position of ZMP changes and there are situations when ZMP reaches the edge of a support polygon. In these situations if additional moments appear, robot will rotate around foot edge and will lose the stability. A way to measure the load on the sole is to analyze force sensors on it [3]. The algorithm of ZMP control is quite straightforward. Compute desired ZMP coordinates, measure the error between current ZMP position and reference ZMP value, apply correcting signal. A very important notion is Center of Pressure (CoP). As the pressure between foot and the ground can be replaced with the force applied in CoP [3]. With this we can define stability condition as ZMP and CoP coincidence. Usually ZMP is required to be under the center of the foot during a single support phase and moves to the other foot in the double support phase.
ASIMO - a bipedal robot of Honda company, was build on top of this theory and the history of its evolution is described in [19]. Nowadays ASIMO is one of the most developed robots, it can interact with humans and perform different task from playing footboll to stair climbing.

In [20] the problem of foot rotation was formulated. The author introduced the Foot-Rotation Indicator (FRI). FRI is the point, that can leave the support polygon. It describes the impeding rotation. When FRI lies outside the support polygon it means that there is rotational moment. In order to control instability of the gait acting on the foot is necessary to compensate this rotation.

2.2 Central Pattern Generator Approach

During the grows humans train the mechanism that can solve some simple problems without mental activities. This mechanism applies trained pattern to the problem. The name of this mechanism is reflex. An experiment conducted in [21] shows that reflexes are significant for locomotion. During human brain reverse engineering a neural network that controls locomotion was found. It was situated in spinal regions which is the reason to name this network Central Pattern Generator (CPG). Another definition of CPG can be found in [22]: CPG is a rhythm generating network in the nervous system which creates and controls the rhythmic motor patterns of animal motions. CPG approach for gait generation was considered in [23]. Neural oscillator was used for generation of motions for bipedal robot. Moreover it can be described as a neural network produces rhythmic pattern outputs without the need for patterned input [6]. CPG is a group of oscillating neurons with some phase between them which results into rhythmic body motion. The pattern is generated not only by the internal neural system but also by the external sensor information, e.g. info from sonar or camera [23]. The key element in CPG is an adaptive neural element that can be described by a pair of first order differential equations. Each pair of the following equations forms an adaptive neuron.

$$\tau_1 \dot{x}_1 = -x_1 - \beta f(\nu_1) - \gamma f(x_2) + u_0 + u_{f_1} \tag{2.11}$$

$$\tau_2 \dot{\nu}_1 = -\nu_1 + f(x_1) \tag{2.12}$$

$$\tau_1 \dot{x}_2 = -x_2 - \beta f(\nu_2) - \gamma f(x_1) + u_0 + u_{f_2} \tag{2.13}$$

$$\tau_2 \dot{\nu_2} = -\nu_2 + f(x_2) \tag{2.14}$$

where x_1 is initial state of neuron, fired by the constant input u_0, x_1, x_2, ν_1 and ν_2 are state variables, dot represents time derivative. When firing reaches some threshold, the neuron goes to state ν_1 through eq. (2.12). When it exceeds some threshold it starts to return to the state x_1 through the eq. (2.11) by the factor β. β represents the rate of adaptations, if β is big, the return to state x_1 is fast. u_{f_1} and u_{f_2} are the feedback inputs from sensors, γ is the coefficient between state variables and f(x) is a threshold function that has zero gain until x is less than threshold value and not zero gain otherwise . Such element works the following way: when it takes constant input it reacts, then adapts and stop reacting. Oscillations generation is the process of reaction and adaptation itself.

Human like gait was achieved by a robot with small Degree of Freedom (DoF) in [23]. The idea of this oscillator was to build a robust system that performs simple control with minimum structure of neural architecture that can be perspective for high DoF robot.

A detailed examination of a biological CPG from an engineering perspective was conducted in [24] but, for most applications, the neuron pair is still approximated with a pair of differential equations [6].

These oscillators can be divided into two types : Simple Sinusoidal Oscillator and Systems of Differential Equations [6], which are described in details bellow in Sec. 2.2.1 and Sec. 2.2.2

2.2.1 Simple Sinusoidal Oscillator

For a sinusoidal signal it is very easy to maintain phase relationships. Human sized bipedal robot controlled by Simple Sinusoidal Oscillator was considered in [25]. The sinusoidal pattern is used for control of parameters, like phase or joint angle. The equation (2.15) shows the example of hip angle control by phase ϕ and hip amplitude A_{hip}.

$$\theta_{hip}(\phi) = A_{hip} sin(\phi) \tag{2.15}$$

However it is a very simple model which is good for systems that are statically stable, but is not appropriate for anthropomorphic robots [6]. Statically stable systems are the systems that van maintain their stability during the continuous absence of movements.is good for systems that are statically stable

2.2.2 Systems of Differential Equations

Analysis of biological CPGs has identified that oscillators are made from pairs of mutually inhibiting neurons [26]. These oscillators are applicable for the generate different gaits and produce various solutions from sinusoidal to more complex forms. It can control anthropomorphic robot even on the rough terrain.

Matsuoka oscillator

Matsuoka oscillator is capable for different gaits and is widely used [6]. They are popular because of their dynamics and in particular limit cycle behavior [27].

The model of neuron with adaptation can be represented in the following way:

$$\begin{aligned} \tau\dot{x} + x &= \sum_{j=1}^{n} c_j s_j - bx' \\ T\dot{x}' + x' &= y \\ y &= g(x - \theta) \end{aligned} \tag{2.16}$$

Here τ is a variable, that represents time delay, x is a membrane potential of the neuron, s_j is an impulse rate of input stimuli, θ is a threshold value below which neuron doesn't fire, c_j are weights of synaptic conjunctions (> 0 for activating synapses and < 0 for inhibitory synapses), y is a firing rate of the neuron, x' is the variable that represents the degree of the adaptation, T (>0) and b(≥ 0) are parameters that specify the time course of the adaptation [27]. Activating synapses are neurons with positive signs of output signal. Inhibitory synapses are neurons that have negative sign of output signal. Degree of the adaptation is the speed of reaction to disturbance. Firing is the notion of process during which the neuron emits an action potential.

Matsuoka in [27] discusses oscillations generated by mutual inhibition between n neurons with adaptation:

$$\begin{aligned} \dot{x}_i + x_i &= -\sum_{j=1}^{n} a_{ij} y_j - bx' + s_i \\ T\dot{x}_i' + x_i' &= y_i \\ y = g(x_i) &\quad (i = 1, ..., n) \end{aligned} \tag{2.17}$$

where a_{ij} represents the strength of inhibitory connection between the neurons. $a_{ij} > 0$ for i≠j and $a_{ij} = 0$ for i = j. $\sum_{j=1}^{n} a_{ij} y_j$ is the total input from the neurons inside a neural network and s_i is the total input from the outside of the network, y_j is input from neuron j.

Matsuoka oscillator is schematically represented in fig. 2.4.

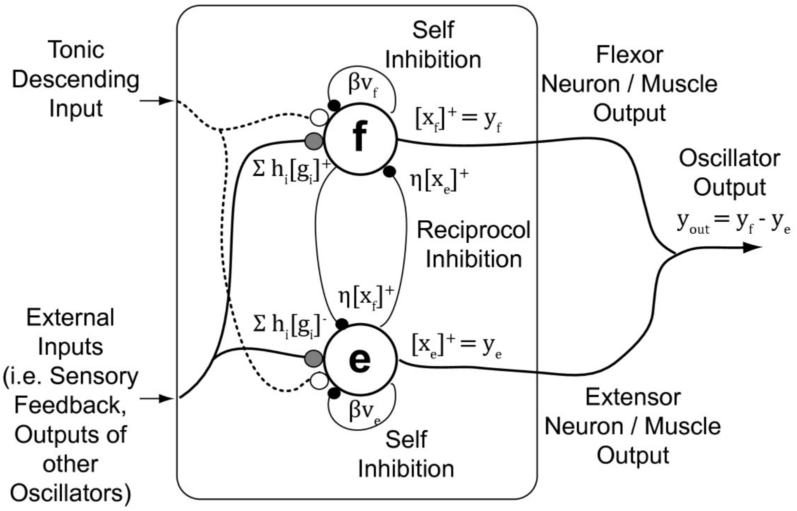

Figure 2.4: Matsuoka oscillator [28]

Two neurons, a flexor (f) and an extensor (e), reciprocally inhibit each other. External inputs (g_i) such as sensory feedback or inputs for other neurons can be either inhibitory or excitatory, depending on the gain (h_i) - that is defined by external sensory inputs. Black circles indicate inhibitory inputs. White circles indicate excitatory inputs. Gray circles can be either inhibitory or excitatory.

Van der Pol oscillator

Van der Pol oscillator benefits are stable limit cycles and relatively interpretable coefficients. Frequency, amplitude and shape coefficients can be identified, although they are not completely independent. Transition between gaits is reached with a simple change of oscillator parameters.

The equation of Van der Pol oscillator is:

$$\ddot{x} - \epsilon(1 - x^2)\dot{x} + x = 0 \tag{2.18}$$

where x is the dynamical variable and $\epsilon > 0$ a parameter, a dot represents time derivative. The van der Pol oscillator has exactly one limit cycle and all other trajectories spiral into it which is proved in [29].

Rayleigh oscillator

Rayleigh oscillator is similar to the Van der Pol oscillator but it has reduced sensory requirements, that is an advantage over conventional prosthetic systems.

The equation of Rayleigh oscillator is:

$$\ddot{x} + \epsilon(3x^2 - 1)\dot{x} + (\omega^2 - \epsilon)x = 0 \tag{2.19}$$

where x is the dynamical variable and $\epsilon > 0$ a parameter, a dot represents time derivative. Comparison of 2.19 and 2.18 shows that Rayleigh oscillator is parametrized with ω Van der Pol oscillator.

Spiking Integrate and Fire with Adaption neurons

Spiking Integrate and Fire with Adaption neurons were used in [30] for bipedal locomotion. The CPG has a hierarchical structure of hip timing, knee timing and output patterning. This structure allowed to tune parameters of every of three hierarchical parts, and so to configure walking frequency, gait and joint angles independently.

2.3 Neural Networks Approach

Despite the fact that oscillators can be described as a neural network they do not require any input for their work. Canonical Neural Networks (NN) have input data and output. There are several groups of NN that are used for bipedal robots control. The most popular of them are described bellow.

2.3.1 Feed-Forward Networks

In this networks each neuron has its own connection and transfer function. Feed-Forward Networks are ready for state motion generation. Input data is current kinematics and sensory data. Current kinematics is the kinematic configuration in the given time frame.

Multi-layer Perceptron

Multi-layer perceptron (MLP) is one of such models. They can have three (input, output, hidden) and more layers (more hidden layers). According to [31], two MLPs were used to specify parameters of a bipedal ditch crossing gait trained with Genetic Algorithm (GA). It produced the best solution in terms of stability. It was more stable and efficient than one from a fully analytical approach, and slightly better than the result obtained with a fuzzy logic based method.

Radial Basis Function Network

Radial Basis Function Network is another example of Feed-Forward Networks (FFN). Activation function is usually Gaussian (2.20) or Euclidean which is described in [32].

$$f(x) = aexp\left(-\frac{(x-b)^2}{2c^2}\right) \tag{2.20}$$

Neurons in hidden layers are connected only with a small part of input neurons. For this reason two types of vectors are necessary: wights vector and centre vector (centre vector dimension is the same as input one). Each vector responds if it is close to the weights vector. Output neurons are functions of linear combinations of network outputs and their wighted connections. It was successfully used for hexapod locomotion [33].

Cerebellar Model Articulation Controller

A Cerebellar Model Articulation Controller (CMAC) is a type of associative memory network based on cerebellum [34]. Input space is continuous and divided into hyper-rectangles. So input should be located in one rectangle at each moment. Numerous hidden layers are stored in slightly moved from the input rectangles. Hence rectangles will overlap on different layers. The output for each layer is a weighted sum of activated rectangles. Activated rectangle is one that are placed near the input point. The CMACs successfully learned the movement patterns and showed resilience to perturbations (including uneven or slippery floors), thereby translating a rigid analytical solution into an adaptable one [35].

2.3.2 Recurrent Networks

The structure of such networks is more complicated that FFN. Thus we can produce more complex patterns and handle different types of input.

For recurrent networks there are different types of input and the difference from FFN is that pattern is not required to be provided as an input but it is generated , tuned or chosen internally by the inputs.

Jordan and Elman Recurrent Neural Networks

Jordan recurrent NN is similar to FFN, the difference is that output of the network is recurrently fed to the input.

The example of Elman recurrent NN is shown in (fig. 2.5). The difference between Elman recurrent NN and Jordan recurrent NN is that delayed output of hidden layer y_i is connected with the inputs of a hidden layer. A delay here is done by u_i nodes from the hidden layer. The Elman network was used in [36], the resilient back-propagation training algorithm was used and the system showed an ability to learn supervised trajectories and interpolate between them.

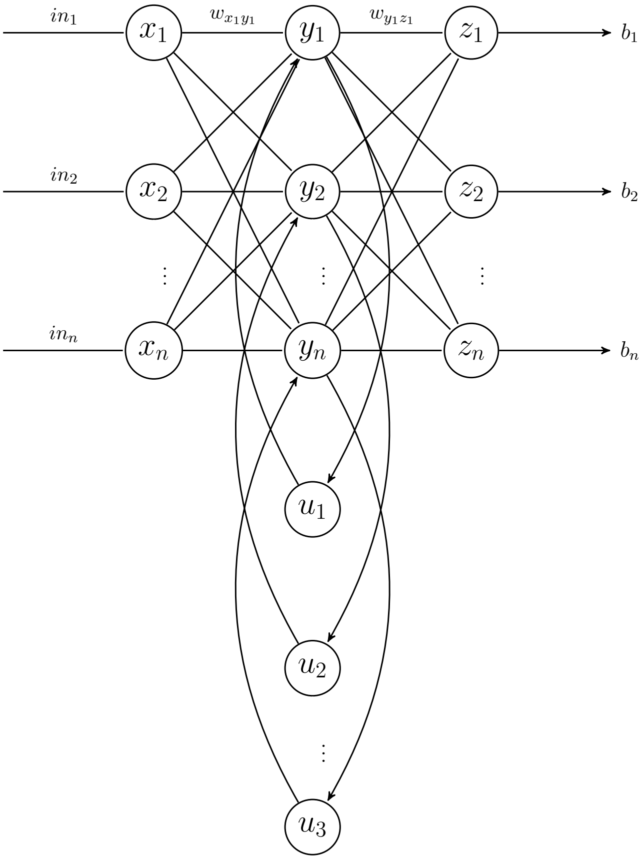

Figure 2.5: Elman recurrent NN. Inputs are x_i, hidden layer contains nodes u_i that delays connected to them outputs of y_i. Outputs of the network are nodes Z_i

Fully Connected Recurrent Neural Networks

In several works [37], [38] a network of several fully connected leaky-integrator neurons were used as a pattern generator.

The network was able to learn multiple sinusoidal patterns, selectable by an input level, but evolution took more generations as the number of required patterns increased. Learned patterns were resilient to noise at the inputs, and even if the input was disrupted by a large amount, the output would quickly return to normal when the input was corrected. It was found that a six neuron model was sufficient to store eight different sinusoidal patterns, as long as the form of those patterns was different [6].

The problem of GA parameters optimization is that for complex systems a huge amount of generations will be necessary. So it can be applied only for offline patterns generation.

Reservoir

Usually recurrent NN should be as small as possible in order to simplify parameters tuning. Reservoir network follows a different principle. It uses a big network with potentially much redundancy. The idea is that during training only output layer is trained and reservoir is unchanged. It works with the following assumption: system dynamics is already present when the network is constructed randomly and the only necessary thing is to adjust weights for it. Thus the training process is very simple, e.g. a linear regression.

Generating trajectories with reservoir can be done taken the human captured data as input [39].

2.4 Hidden Markov Model Approach

Learning by imitation is one of commonly used approaches for bipedal locomotion researches. Obviously, direct copying of movements of an operator is impossible due to different kinematic and dynamic characteristics. So, from the operator we can take only the pattern and then imitate it on the robot. Observing here is estimating the underlying state variables. It is based on the source system through the output data. For this task Hidden Markov Models (HMM) can be used [40].

A recognition algorithm is used for identification of movement. If movement was not identified, it is added to the database as new. Hidden Markov model provides abstraction of the pattern which is used for motion synthesis. Although typically used for imitation tasks, HMM have also been employed to detect problematic states. HMM itself is a finite state automaton. It is defined by a set of states with initial and final states, transition probability matrix where element a_{ij} is the probability of transition from state i to state j.

Controller basing on its input generates an output signal that is called the control signal. Input for controller is difference between reference signal and system output. In order to obtain a finite set of decision patterns, it is necessary to divide the control space into a finite number of subspaces. According to [41], the following steps are necessary to define an HMM-based controller:

1. A correspondence between the control signal and controller input (error signal) can be established due to the definition of HMM the control signal depends only on a

finite number of previous input signals.

2. Define a set of patterns. At each time the control signal belongs to one of the patterns. Also control signal at each moment of time corresponds to one of the set of k previous input signals.
3. It is necessary to label the data. It means that the set of input signals is mapped to the set of possible control signals.
4. Train the model by the data describing control signals.
5. Collect a set of trained models.

HMM are good for simple systems control, however it requires a lot of space and time for teaching the HMM controller to work with complex multi body systems.

2.5 Rule Based Approach

After the identification of the current state we can find the next one using rule-based method. For bipedal robots fuzzy-logic systems were used in some researches [42]. A fuzzy logic controller helps to deal with a bit of uncertainty in the environment. And it can be used to vary ZMP position in the support polygon [42]. For the work it requires optimization of rules, that were basically chosen [31]. The main idea is to divide the set of all possible system configurations into the clusters. For each cluster assign the control function. Hence during the work control function will be chosen according to the current robot configuration from the set of all possible control functions according to the current configuration. Fuzzy logic controller is a perspective approach for solving dynamical stability problem for bipedal robots because it divide all the configuration space into the subspaces. For each subspace control function is defined in an optimal way.

Chapter 3

Bipedal Locomotion Control

According to the literature review there are a lot of different approaches for bipedal robot locomotion. In this work several of them were combined. We define two support phases of walking: single support phase (during this phase only one foot is in contact with the ground) and double support phase (two feet are in the contact ground simultaneously).

3.1 Locomotion Phases

We can break bipedal locomotion into the following phases [43]:

1. Stance phase.

   During this phase the load is on a bearing leg. It starts from the double support phase during weight acceptance. Than during single limb support phase only bearing leg is in the contact with the ground.

2. Swing phase.

   During this phase there is limb advancement. It starts from double support phase and than finishes with single support phase.

This phases can be identified on (fig. 3.1).

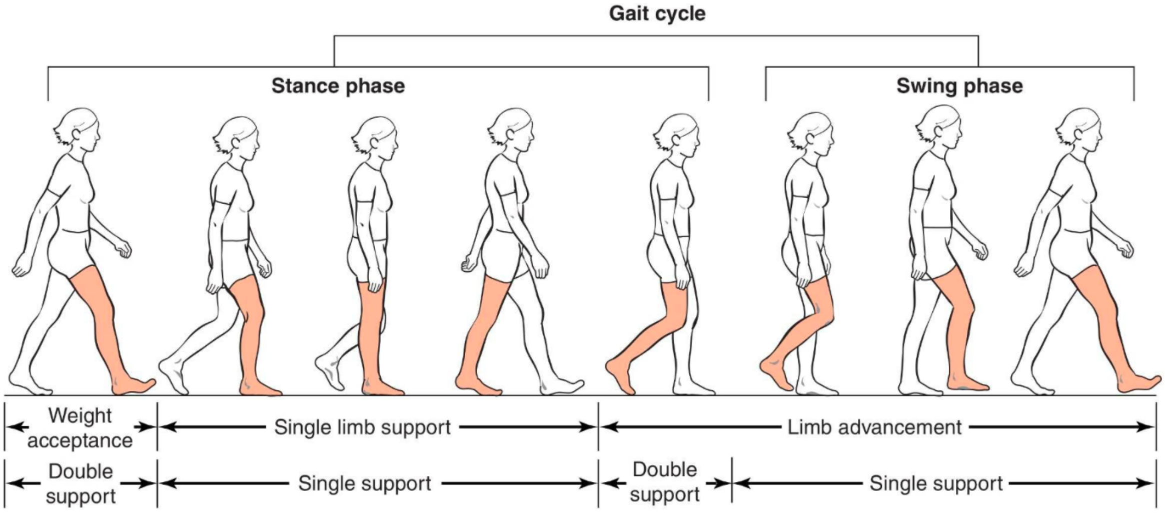

Figure 3.1: Human gait cycle [44]

3.2 Human Locomotion Models

One of the most simple models of human during single support phase is cart-table model, which is represented in fig. 3.2.

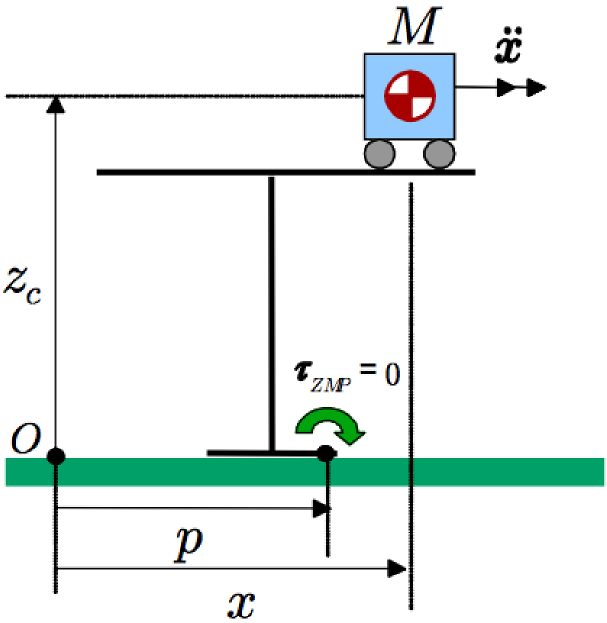

Figure 3.2: Cart-table model [14]. M is cart mass. Z_c is the height of cart CoM. τ_{ZMP} is the moment around ZMP. x is the cart trajectory, $\ddot{x}$ is the cart acceleration

When cart is far away from the Center of Mass (CoM), ZMP point goes out of support area shown in fig. 3.2. When it happens the system becomes unstable. The distance from zero point O to the center of table support polygon is denoted as X, the distance from zero to ZMP coordinate is denoted as P. The equation that describes this process was derived in [14] and is represented bellow:

$$P = x - \frac{Z_c}{g}\ddot{x} \tag{3.1}$$

where Z_c is the height of CoM of the cart, g is the gravitational acceleration and $\ddot{X}$ is the second time derivative.

In order to prevent system falling we have to control the cart in the way that maintains the trajectory of ZMP of the system which guaranties that ZMP always stays inside the support polygon. When ZMP lies within the support polygon it maintains the dynamical stability of the system. When it leaves the region of support polygon we have a risk that the system can loose the stability and fall. Thus our target is to define the desired ZMP trajectory of the system. We define two planes in which the robot can move: frontal and sagittal planes. Also while there exists a transverse plane as well. We now consider only locomotion in frontal plane for simplicity. All the planes are represented in fig. 3.3. Desired trajectory for our system in frontal plane is provided in fig. 3.4.

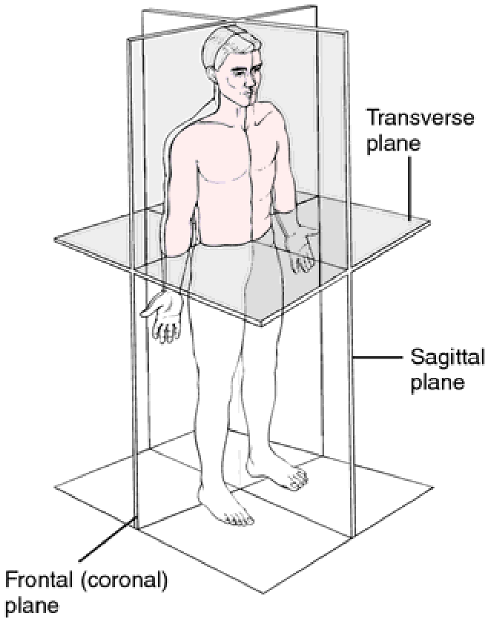

Figure 3.3: Human planes [45]

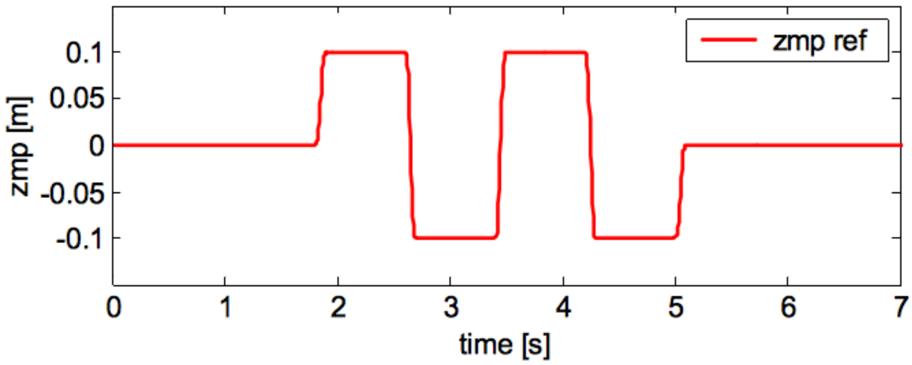

Figure 3.4: Desired ZMP trajectory in frontal plane [14]. ZMP reference is the desired ZMP trajectory. ZMP is measured from zero in frontal plane.

Desired trajectory represents the movement of ZMP from the region of standing foot to the region of swinging foot in the moment of the step completing. Here the amplitude of frictions is defined by parameters of robot (step width in frontal plane). Thus we can build a simple control system represented in fig. 3.5. The input of this system is the reference trajectory of ZMP. Controller takes the error between reference ZMP trajectory and real measured ZMP trajectory as an input and generates control signal that is applied to the cart.

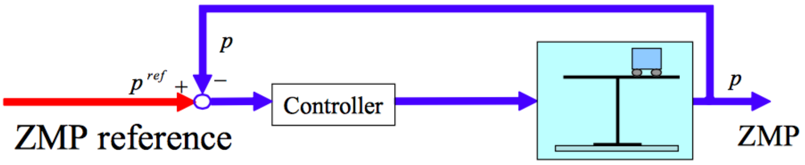

Figure 3.5: PID regulated ZMP control [14]

If the controller is PID (proportional integral derivative) the regulator which maintains ZMP trajectory close to the desired one. We can see in in fig. 3.6 that our real ZMP trajectory follow the desired trajectory with the late. The result of such control is represented in fig. 3.6.

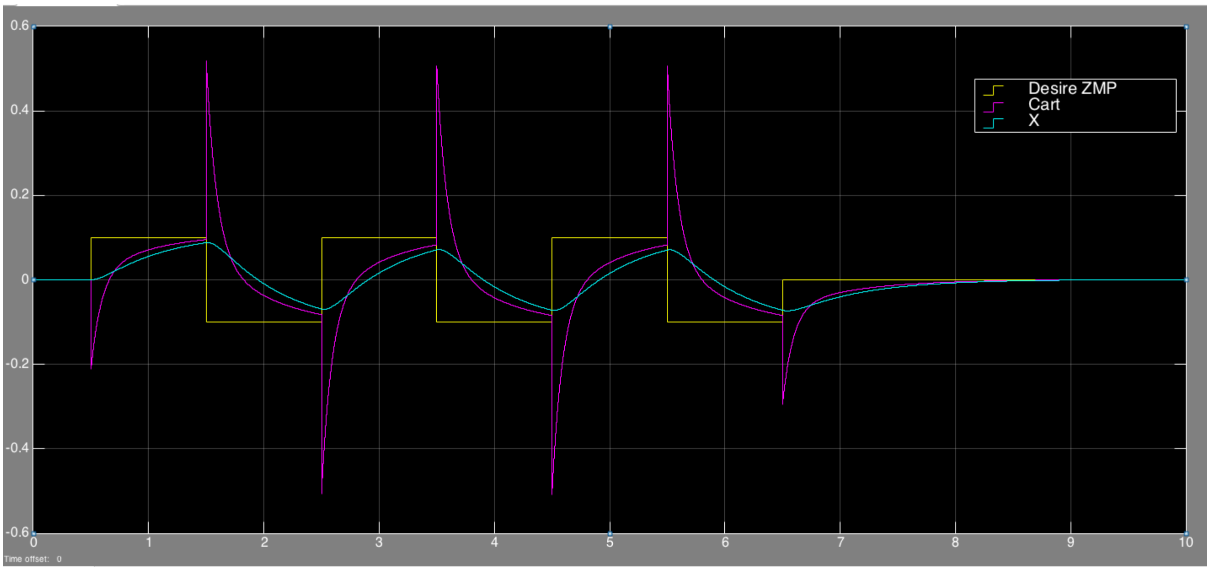

Figure 3.6: PID regulated ZMP trajectories. Due to fast change of desired trajectory in the moment of phase change real ZMP trajectory the fast risings.

Kajita et all in [14] introduced a simple idea, to look forward to the desired ZMP and try to predict what will happen before it happens. Hence the theory of optimal control comes into view and it will be described bellow how to make a predictive control system for such an unstable object as cart on the table model. Further Chapter 4 presents such controller being applied to the model of a biped robot and Chapter 5 demonstrates the evaluation of it.

3.3 Optimal Control

Next we introduce a discrete-time state-space model. Equation 3.2 introduces this model. This system of equations represents a dependency between the next state and current state of system though variables x and y [46].

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{3.2}$$

Where k is a discrete time index, x(k) is a vector of state values (trajectory, velocity and acceleration), u(k) is a control signal (jerk), y(k) is a vector of system output values (trajectory, velocity and acceleration). A, B, C, D are appropriately dimensioned real matrices that defines system parameters and would be defined further in this chapter [46].

We define G, the transfer function associated with model defined be eq. (3.2). G is defined as in 3.3

$$G(Z) = C(ZI - A)^{-1}B + D \tag{3.3}$$

Where Z is Z-transform variable and I is an identity matrix.

We can estimate the size of the transfer function by measures H_2 and H_∞. H_2 norm will be denoted as $||G(Z)||_2$ and is defined as in the following way:

$$||G(Z)||_2 = Tr\{B^TXB + D^TD\} \tag{3.4}$$

Where X should satisfy eq. (3.5):

$$X = A^TXA + C^TC \tag{3.5}$$

H_2 norm of the transfer function is a gain in power from input to output, assuming that the input signal is white [46].

H_∞ is denoted as $||G(Z)||_\infty$ and is defined as:

$$||G(Z)||_\infty = \sup_\omega \frac{||z||_2}{||\omega||_2} \tag{3.6}$$

where $z = G\omega$, ω is assumed to be a realization of a unit power, Gaussian, white-noise process and z is the real values vector of input. Hence H_∞ defines the maximum possible gain in power from input ot output.

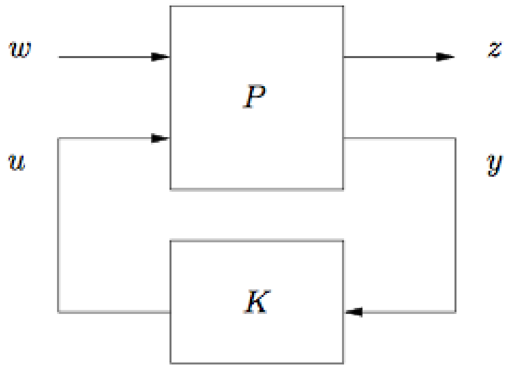

Figure 3.7: Generalized regulator [46]

There is a generalized regulator in fig. 3.7. The input for the system is ω. The system is controlled by input signal u, ω represents all exogenous inputs including noise and disturbances. The system returns an output signal z, while y is measurements and references output. Thus K here is a feedback controller and P represents the controlled system.

The following matrix equation

$$X = A^T X A + Q - \bar{L}^T \bar{R}^{-1} \bar{L} \tag{3.7}$$

is known as Discrete Algebraic Riccati Equation (DARE). Here $\bar{R} = R + B^T X B$ and $\bar{L} = L + B^T X A$. The solution of this equation derived in [46]. Also we can write DARE in a short form: $X = D(A, B, Q, R, L; X)$ and $D(A, B, Q, R, L; X) = A^T X A + Q - \bar{L}^T \bar{R}^{-1} \bar{L}$.

3.3.1 Preview Control

Katayama et al. in [47] introduced a new method of control for general regulators with preview control. The key idea was to look forward for N discrete steps and predict the desirable value of controlled signal. The form of the control signal was derived as a theorem in [47]. The theorem says that the optimal control signal have the form of:

$$u(k) = -G_I \sum_{i=0}^{k} e(i) - G_x x(k) - \sum_{l=1}^{N} G_d(l) y_d(k+l) \tag{3.8}$$

Where $G_I = [R + \tilde{B}^T \tilde{K} \tilde{B}]^{-1} \tilde{B}^T \tilde{K} \tilde{I}$ represents integral coefficient, e(i) is an error between controlled value and its referenced value, $G_x = [R + \tilde{B}^T \tilde{K} \tilde{B}]^{-1} \tilde{B}^T \tilde{K} \tilde{F}$ represents proportional coefficient, x(k) is the controlled value, $G_d(1) = -G_I$ and $G_d(l) = [R + \tilde{B}^T\tilde{K}\tilde{B}]^{-1}\tilde{B}^T\tilde{X}(l-1)$ represents feature coefficients at time l, $y_d(t)$ is a value of reference signal at time t.

with the solution of DARE $\tilde{K}$ in the form:

$$\tilde{K} = \tilde{A}^T\tilde{K}\tilde{A} - \tilde{A}^T\tilde{K}\tilde{B}[R + \tilde{B}^T\tilde{K}\tilde{B}]^{-1}\tilde{B}^T\tilde{K}\tilde{A} + \tilde{Q} \tag{3.9}$$

where:

$$\tilde{B} = \begin{bmatrix} CB \\ B \end{bmatrix}, \tilde{I} = \begin{bmatrix} I_p \\ 0 \end{bmatrix}, \tilde{F} = \begin{bmatrix} CA \\ A \end{bmatrix}, \tilde{Q} = \begin{bmatrix} Q_e & 0 \\ 0 & Q_x \end{bmatrix}, \tilde{A} = \begin{bmatrix} \tilde{I} & \tilde{F} \end{bmatrix} \tag{3.10}$$

And finally I_p is a $p * p$ identity matrix p is defined by the size of matrices C and A in order to allow computation of matrix $\tilde{A}$.

Kajita et al. in [14] proved that the necessary prediction for the cart table model is 1.6 seconds due to the dependency law on this parameter which is shown in fig. 3.8

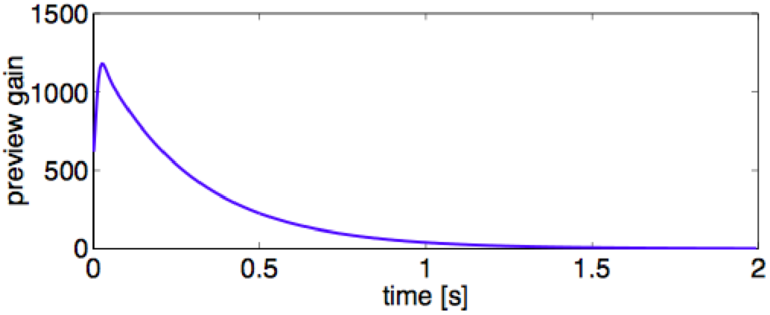

Figure 3.8: Power Gain dependency from the time [14]

For the moment we have everything to design control system for the model of bipedal robot.

Chapter 4

Implementation

The modeling of real-world objects is a very difficult task. It takes into account physical lows, gravity and frictions. Simulation environment for this thesis is Simulink. Simulink is a framework in matlab environment. It provides primitives that can be used for real world models building. Also it provides the interface for high-level frameworks, that realizes high-level primitives for modeling. SimMechanics is one of packages in Simulink. It contains the blocks for mechanical modeling of multi body systems. Robots are multi body systems and hence SimMechanics is a suitable choice for the environment. Initially the cart table model was developed. The design is represented in fig. 4.1

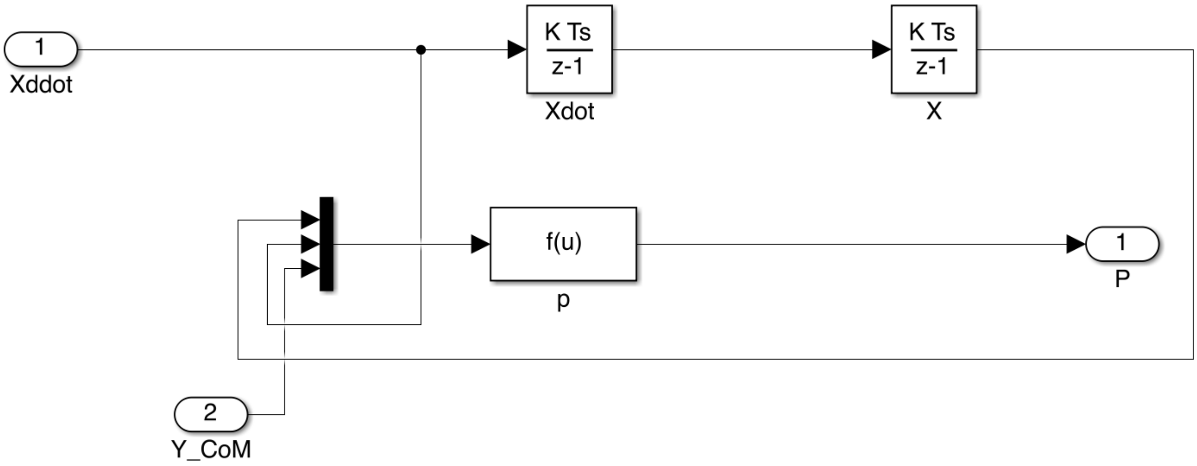

Figure 4.1: Cart-table model. Xddot is acceleration, Xdot is velocity, X is trajectory of cart CoM. Y_CoM is the height of cart CoM. P is computed ZMP position.

Further it will be represented as a block on fig. 4.2.

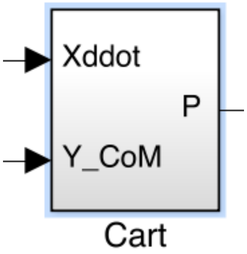

Figure 4.2: Cart-table model block. Xddot is input second time derivative (velocity of CoM). Y_COM is the height of CoM that can deviate and thus cannot be constant. P is measured ZMP.

This system is controlled by the second time derivative that represents speed. For the initial control model Proportional Integral Derivative (PID) regulator was applied. The system of PID regulated cart-table model is shown in fig. 4.3.

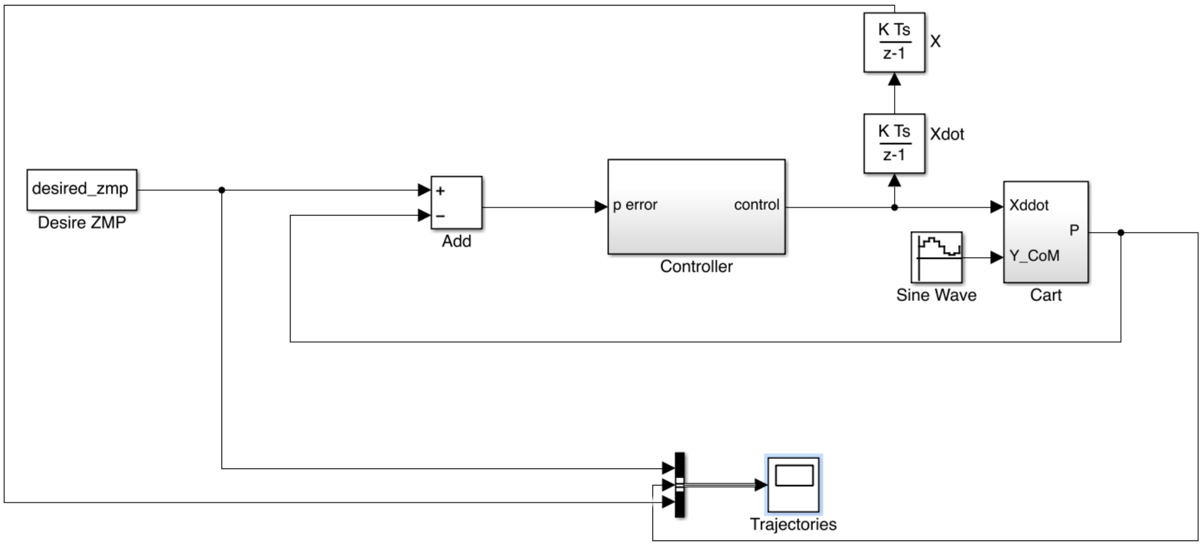

Figure 4.3: PID regulated cart-table. X is trajectory of CoM. Desired ZMP is reference signal of ZMP. P is real computed ZMP.

The regulator itself looks as on fig. 4.4.

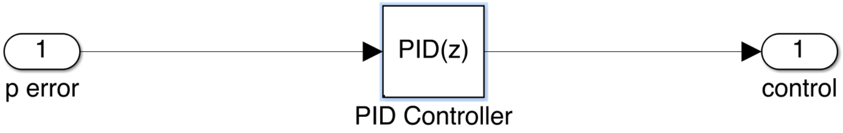

Figure 4.4: PID controller. P error is the error between reference ZMP and measured ZMP. Control is velocity of cart table model that is necessary to compensate the error between desired ZMP and measured ZMP.

The results of such control are represented on fig. 4.5.

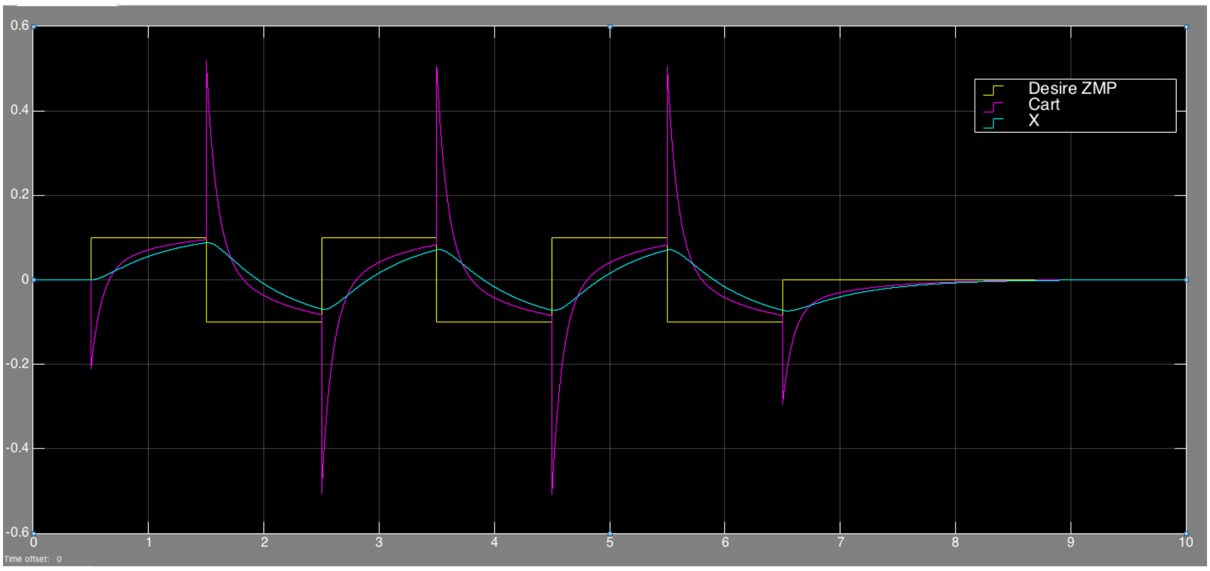

Figure 4.5: PID regulator control results.

We can see that the actual ZMP trajectory lates and due to this fact the CoM trajectory reaches the bound of support polygon and thus it makes the system unstable.

After that the preview control with 1.6 second prediction was applied. The model of preview controller is represented on fig. 4.6.

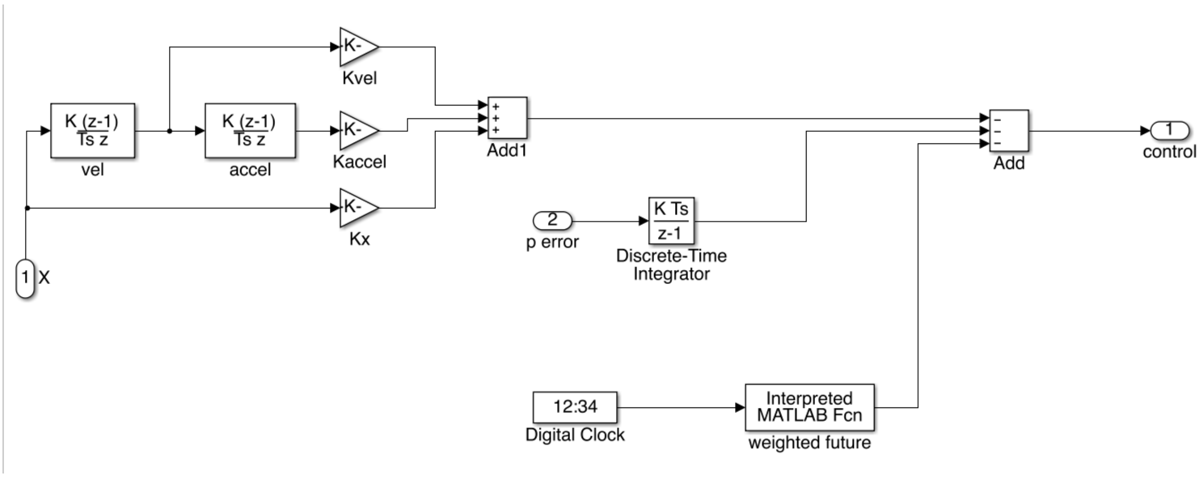

Figure 4.6: Preview controller design. Here X is input of current CoM trajectory, p error is error between reference ZMP and measured ZMP, weighted feature is function that computes weighted sum of 10000 next time samples of desired ZMP. Control is controll signal that represents the third time derivative of trajectory of cart.

The control signal for preview controller is the third time derivative of trajectory and due to this fact a transition block between controller and cart-table model should be provided. This block is an integrator because integration of control signal gives us exactly the second time derivative (acceleration). Preview control of cart table model is represented on fig. 4.7

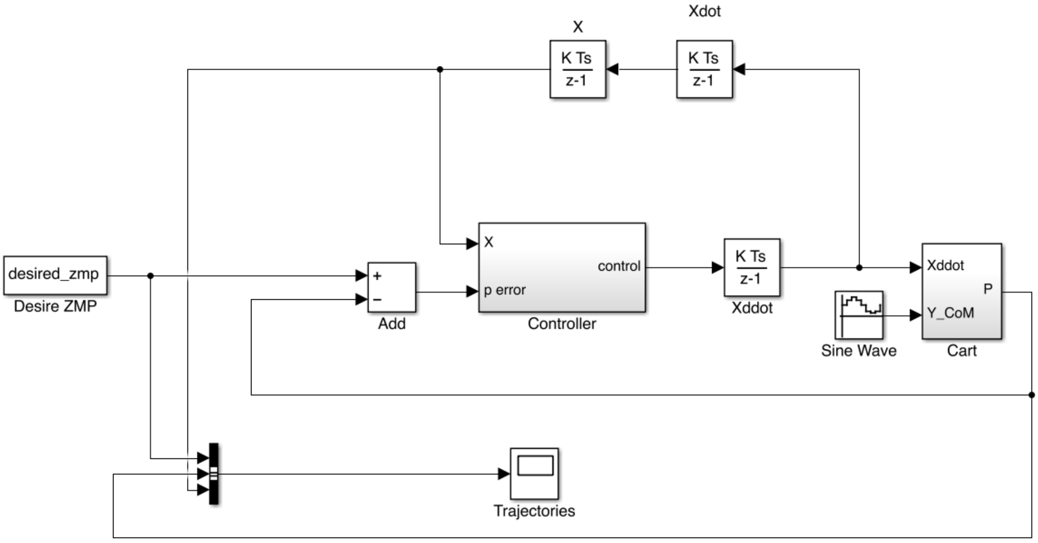

Figure 4.7: Preview control of cart table model.

The results of this controller are represented on fig. 4.8 and are much better : we can see no late response and no overshooting in ZMP trajectory. Thus the system become much more stable than in the PID case.

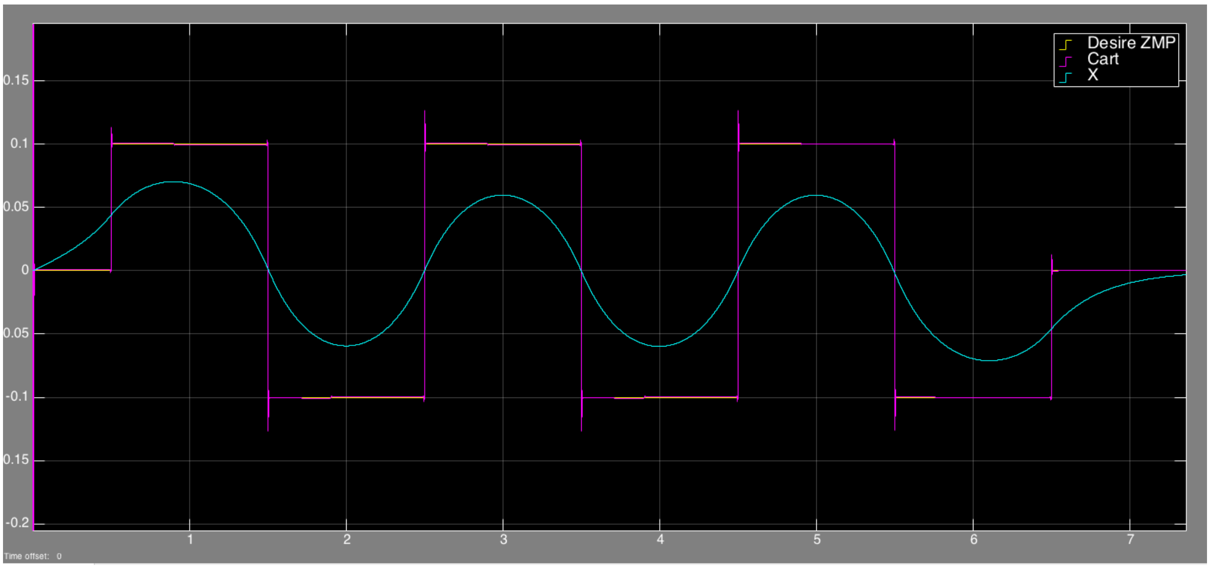

Figure 4.8: Preview controller model

This developed controller should be applied to the model of robot. Initialy we will consider the simple model of robot. The experiment is described in Chapter 5.

Chapter 5

Evaluation

After tuning of preview controller parameters (N samples look ahead) a simple model of bipedal robot with 12 Degrees of Freedom (DoF) was built. The model is represented on fig. 5.1. In [48] a simplification of bipedal model was introduced. According to this simplification we can write ZMP equation in the form of 5.1.

$$\begin{aligned} X_{CoG}(t) &= X_{CoG}(0)cosh(\sqrt{\frac{g}{\alpha Z_{CoG}}}t) + \sqrt{\frac{\alpha Z_{CoG}}{g}}\dot{X}_{CoG}(0)sinh(\sqrt{\frac{g}{\alpha Z_{CoG}}}t) \\ Y_{CoG}(t) &= Y_{CoG}(0)cosh(\sqrt{\frac{g}{\beta Z_{CoG}}}t) + \sqrt{\frac{\beta Z_{CoG}}{g}}\dot{Y}_{CoG}(0)sinh(\sqrt{\frac{g}{\beta Z_{CoG}}}t) \end{aligned} \tag{5.1}$$

Where X_{CoG}, Y_{CoG}, $\dot{X}_{CoG}$ and $\dot{Y}_{CoG}$ are X coordinate, Y coordinate, X velocity and Y velocity of Center of Gravity (CoG) respectively. Parameters α and β defines robot configuration and the optimal values for model on fig. 5.1 are 0.65 and 0.45 respectively.

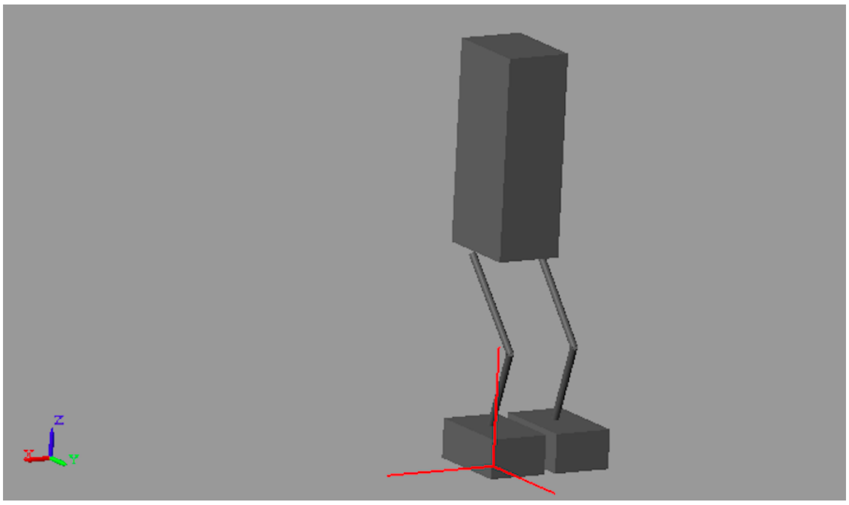

Figure 5.1: 12 DoF bipedal robot model

Analytical solution of 3.1 applied to bipedal robot with provided α and β under assumptions from [48] can be defined as follows:

$$X_{CoM} = C_1 e^{-\omega t} + C_2 e^{\omega t} \tag{5.2}$$

Where $C_1 = \dfrac{0.2(1+e^{0.5\omega})}{(e^{-0.5\omega} - e^{0.5\omega})}$, $C_2 = -0.2 - C_1$, $\omega = \sqrt{\dfrac{g}{Z_c}}$.

The trajectory of CoM in frontal plane of this analytical solution is represented on fig. 5.2.

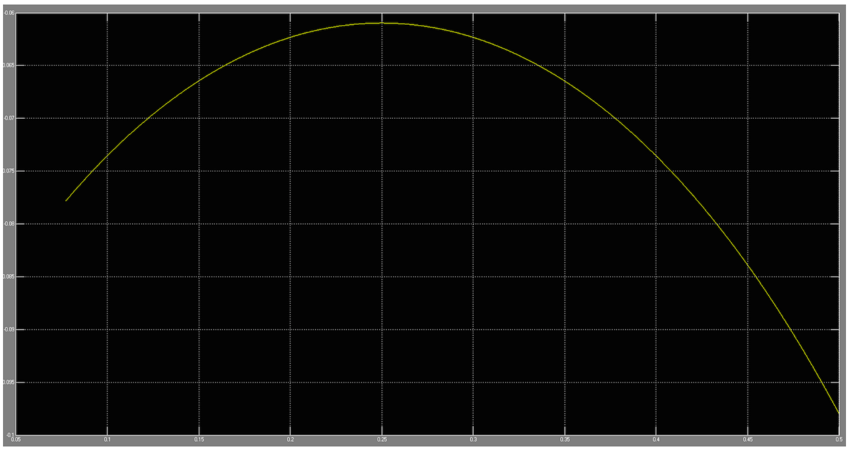

Figure 5.2: Analytically synthesized CoM trajectory.

A direct application of preview controller shows perfect results which are represented on fig. 5.3. However these results cannot be used on practice because motors cannot work with given acceleration. Thus the preview controller was restricted with some values, that represents maximum motors acceleration. Considering the preview control application for

CoM trajectory generation in frontal plane it is necessary to mention that some related works, for example [49] consider control signal not in the form of 3.8 but in the form of 5.3.

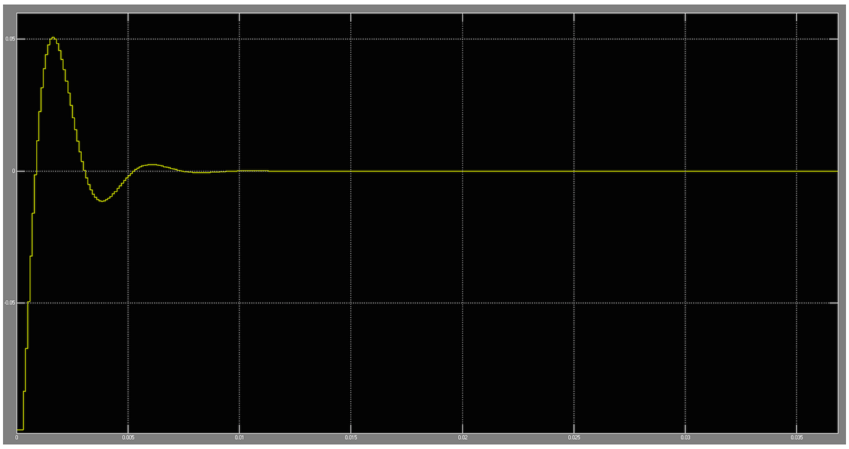

Figure 5.3: Preview control generated CoM trajectory without restriction.

$$u(k) = -G_I e(i) - G_x x(k) - \sum_{l=1}^{N_l} G_d(l) y_d(k+l) \tag{5.3}$$

There is motivation for this form mentioned in articles, but it is obvious that for small disturbances integrated error and current error are small and close to zero. Preview controllers of two forms 3.8 and 5.3 were applied with restrictions to the robot model to generate CoM trajectory in frontal plane. Results are provided bellow.

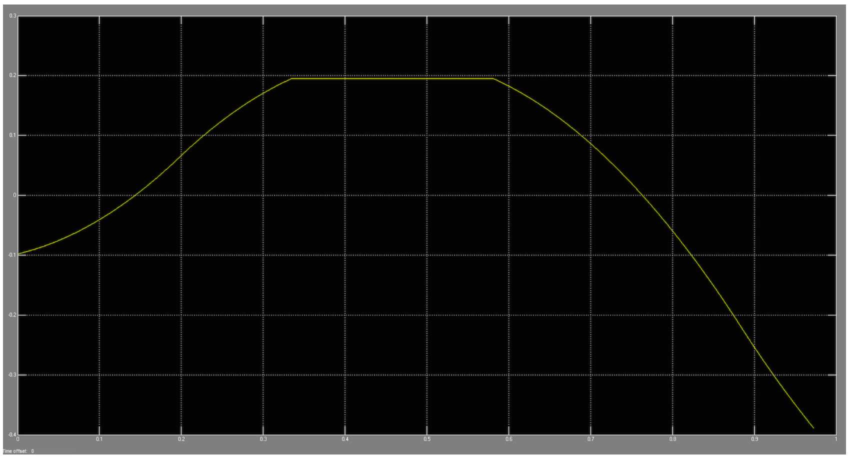

Figure 5.4: CoM Trajectory in frontal plane generated by controller described by 5.3

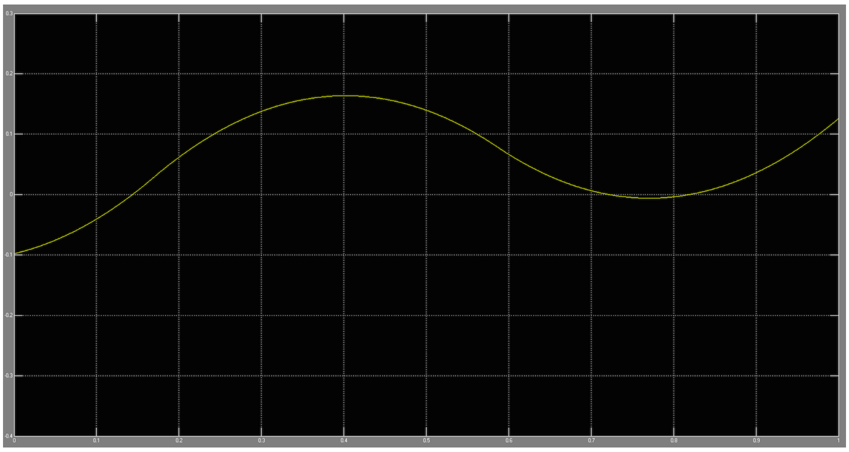

Figure 5.5: CoM Trajectory in frontal plane generated by controller described by 3.8

We can see that trajectory on fig. 5.5 is smoother than trajectory on fig. 5.4. It means that there would be less disturbances provided by motors. Thus the form 3.8 is more preferable than the form 5.3. Moreover we can see that the trajectory of analytical derivations is close to the trajectory of preview controller. It means that the preview controller can efficiently solve the problem of trajectory generation. This approach seems to be more preferable because it allows a robot to operate in environments with uncertainty.

Chapter 6

Future Work

Due to the limitations of the model it is necessary to develop a model that is more similar to a real robot. Moreover to apply the preview controller on a real robot and even on advanced simulator it is still necessary to develop an inverse kinematics module that will transform CoM trajectories to trajectories for every joint. Here a rule based approaches can be used because the robot is a redundant construction and for every trajectory there will be a lot of configurations and thus it is necessary to choose the best one. Moreover it is necessary to combine two controllers in sagital and frontal planes. The next step after tuning control parameters on the model is to apply it to a real robot. Described in this thesis control method is capable for pattern generation and so future work should consider optimization methods for generated trajectory to prevent the robot from falling on the rough terrain. When control algorithm is developed the structure of the robot should be revised in order to maintain optimal criteria. For example this criteria could be an energy consumption or the speed of the movements. After several iterations of algorithm and structure optimization the robot will become able for walk, however further investigations of AI and object detection are still necessary. It is reasoned by the level of autonomy that we want robot to achieve. It is impossible to program all possible conditions for robot and thus it have to deal with some level of uncertainty.

Chapter 7

Summary

In this work different approaches to bipedal locomotion were considered. Classification of existing methods was also done and it was the basics for development of a dynamic balance algorithm for bipedal robot locomotion. Two approaches were chosen: analytical one (ZMP criteria for dynamical stability) and CPG (controller output should generate a pattern for locomotion). Results of implemented preview controller on provided models show stable smooth trajectories and thus it seems reliable to try this approach on the real robot. There are works [14, 50] that already applied preview controller to real robots. However in their works the authors don't consider application of these techniques to simulation of the robot. As shown in Chapter 5 a direct application of preview controller to robot model gives very good but physically unreachable results. Due to the fact that motors have limited accelerations, the system cannot react as fast as the controller tries to stabilize the system. However the controller was limited by physical constraints and the results of control became similar to the fully analytical solution during one step period. Thus the developed approach is promising because it can adapt to disturbances and uneven terrain despite the fact that it produces only the pattern for the robot. Thus it was developed pattern generation subsystem that can be applied to a real robot. It requires additional control to maintain another stability criteria, for example FRI. Moreover stable bipedal locomotion requires efficient processing of extrinsic sensors data in order to predict possible disturbances before they are applied to the robot.

Bibliography

[1] Rolf Pfeifer, Max Lungarella, and Fumiya Iida. Self-organization, embodiment, and biologically inspired robotics. *science*, 318(5853):1088–1093, 2007.

[2] Tomomichi Sugihara and Yoshihiko Nakamura. A fast online gait planning with boundary condition relaxation for humanoid robots. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 305–310. IEEE, 2005.

[3] Miomir Vukobratović and Branislav Borovac. Zero-moment pointthirty five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173, 2004.

[4] Yariv Bachar. Developing controllers for biped humanoid locomotion @ONLINE, may 2015. URL `http://www.inf.ed.ac.uk/publications/thesis/online/IM040191.pdf`.

[5] Kemalettin Erbatur, Akihiro Okazaki, Keisuke Obiya, Taro Takahashi, and Atsuo Kawamura. A study on the zero moment point measurement for biped walking robots. In *Advanced Motion Control, 2002. 7th International Workshop on*, pages 431–436. IEEE, 2002.

[6] Joe Wright and Ivan Jordanov. Intelligent approaches in locomotion-a review. *Journal of Intelligent & Robotic Systems*, pages 1–23, 2014.

[7] Zhe Tang, Meng Joo Er, and C-J Chien. Analysis of human gait using an inverted pendulum model. In *Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence). IEEE International Conference on*, pages 1174–1178. IEEE, 2008.

[8] David A Winter. Human balance and posture control during standing and walking. *Gait & posture*, 3(4):193–214, 1995.

[9] Adam C Knight and Wendi H Weimar. Difference in ratio of evertor to invertor activity between the dominant and nondominant legs during simulated lateral ankle sprain. *Journal of sport rehabilitation*, 22(4), 2013.

[10] Steven Hartley Collins and Andy Ruina. A bipedal walking robot with efficient and human-like gait. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1983–1988. Ieee, 2005.

[11] C Golliday Jr and Hooshang Hemami. An approach to analyzing biped locomotion dynamics and designing robot locomotion controls. *Automatic Control, IEEE Transactions on*, 22(6):963–972, 1977.

[12] Stuart O Anderson, Martijn Wisse, Christopher G Atkeson, Jessica K Hodgins, GJ Zeglin, and Brian Moyer. Powered bipeds based on passive dynamic principles. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 110–116. IEEE, 2005.

[13] Ian R Manchester, Uwe Mettin, Fumiya Iida, and Russ Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, page 3, 2011.

[14] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1620–1626. IEEE, 2003.

[15] Kazuhisa Mitobe, Genci Capi, and Yasuo Nasu. Control of walking robots based on manipulation of the zero moment point. *Robotica*, 18(06):651–657, 2000.

[16] Philippe Sardain and Guy Bessonnet. Forces acting on a biped robot. center of pressure-zero moment point. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 34(5):630–637, 2004.

[17] MHP Dekker. Zero-moment point method for stable biped walking. *Eindhoven University of Technology*, 2009.

[18] Steven H Collins, Martijn Wisse, and Andy Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *The International Journal of Robotics Research*, 20(7):607–615, 2001.

[19] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1321–1326. IEEE, 1998.

[20] Ambarish Goswami. Postural stability of biped robots and the foot-rotation indicator (fri) point. *The International Journal of Robotics Research*, 18(6):523–533, 1999.

[21] S_ Grillner. Locomotion in vertebrates: central mechanisms and reflex interaction. *Physiological reviews*, 55(2):247–304, 1975.

[22] Wai Man Lee and Yeung Yam. Construction of central pattern generator using piecewise affine systems. In *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*, pages 1729–1734. IEEE, 2007.

[23] Seiichi Miyakoshi, Gentaro Taga, Yasuo Kuniyoshi, and Akihiko Nagakubo. Three dimensional bipedal stepping motion using neural oscillators-towards humanoid motion in the real world. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 1, pages 84–89. IEEE, 1998.

[24] Kuanyi Zhu, Dingguo Zhang, and Li Lan. On central pattern generator of biological motor system. In *Control, Automation, Robotics and Vision, 2006. ICARCV'06. 9th International Conference on*, pages 1–5. IEEE, 2006.

[25] Jun Morimoto, Gen Endo, Jun Nakanishi, and Gordon Cheng. A biologically inspired biped locomotion strategy for humanoid robots: Modulation of sinusoidal patterns by a coupled oscillator model. *Robotics, IEEE Transactions on*, 24(1):185–191, 2008.

[26] Sten Grillner, Tanja Deliagina, A El Manira, RH Hill, GN Orlovsky, P Wallén, Ö Ekeberg, and A Lansner. Neural networks that co-ordinate locomotion and body orientation in lamprey. *Trends in neurosciences*, 18(6):270–279, 1995.

[27] Kiyotoshi Matsuoka. Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biological cybernetics*, 52(6):367–376, 1985.

[28] Guang Lei Liu, Maki K Habib, Keigo Watanabe, and Kiyotaka Izumi. Central pattern generators based on matsuoka oscillators for the locomotion of biped robots. *Artificial Life and Robotics*, 12(1-2):264–269, 2008.

[29] Takashi Kanamaru. Van der pol oscillator. *Scholarpedia*, 2(1):2202, 2007.

[30] Alexander Russell, Garrick Orchard, and Ralph Etienne-Cummings. Configuring of spiking central pattern generator networks for bipedal walking using genetic algorthms. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 1525–1528. IEEE, 2007.

[31] Pandu Ranga Vundavilli and Dilip Kumar Pratihar. Dynamically balanced optimal gaits of a ditch-crossing biped robot. *Robotics and Autonomous Systems*, 58(4): 349–361, 2010.

[32] Rod Downey. Euclidean domains and euclidean functions @ONLINE, may 2015. URL `http://homepages.ecs.vuw.ac.nz/~downey/chicago_domain_2010.pdf`.

[33] Winfried Ilg and Karsten Berns. A learning architecture based on reinforcement learning for adaptive control of the walking machine lauron. *Robotics and Autonomous Systems*, 15(4):321–334, 1995.

[34] JS Albus. A new approach to manipulator control: The cerebellar model articulation controller. *Transaction of the ASME Journal of Dynamic Systems*, 97(3):270–277, 1975.

[35] Christophe Sabourin and Olivier Bruneau. Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using cmac neural networks. *Robotics and Autonomous Systems*, 51(2):81–99, 2005.

[36] Karsten Berns, Rüdiger Dillmann, and Stefan Piekenbrock. Neural networks for the control of a six-legged walking machine. *Robotics and autonomous systems*, 14(2): 233–244, 1995.

[37] Torsten Reil and Phil Husbands. Evolution of central pattern generators for bipedal walking in a real-time physics environment. *Evolutionary Computation, IEEE Transactions on*, 6(2):159–168, 2002.

[38] V Petridis and A Papaikonomou. Recurrent neural networks as pattern generators. In *Proc IEEE International Conference on Neural Networks*, pages 872–875. Citeseer, 1994.

[39] Francis Wyffels and Benjamin Schrauwen. Design of a central pattern generator using reservoir computing for learning human motion. In *Advanced Technologies for Enhanced Quality of Life, 2009. AT-EQUAL'09.*, pages 118–122. IEEE, 2009.

[40] Tetsunari Inamura, Iwaki Toshima, Hiroaki Tanie, and Yoshihiko Nakamura. Embodied symbol emergence based on mimesis theory. *The International Journal of Robotics Research*, 23(4-5):363–377, 2004.

[41] Jie Yang and Yangsheng Xu. Hidden markov model for control strategy learning. Technical report, DTIC Document, 1994.

[42] Jong Hyeon Park. Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motions of biped robots. *Fuzzy sets and systems*, 134(1):189–203, 2003.

[43] Mostafa Rostami and Guy Bessonnet. Impactless sagittal gait of a biped robot during the single support phase. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1385–1391. IEEE, 1998.

[44] P. Hill. Gait @ONLINE, may 2015. URL `https://www.studyblue.com/notes/note/n/gait/deck/13079492`.

[45] American heritage stedman's medical dictionary @ONLINE, may 2015. URL `http://medical-dictionary.thefreedictionary.com`.

[46] Andrew Hazell. Discrete-time optimal preview control. 2008.

[47] Tohru Katayama, Takahira Ohki, Toshio Inoue, and Tomoyuki Kato. Design of an optimal controller for a discrete-time system subject to previewable demand. *International Journal of Control*, 41(3):677–699, 1985.

[48] Taesin Ha and Chong-Ho Choi. An effective trajectory generation method for bipedal walking. *Robotics and Autonomous Systems*, 55(10):795–810, 2007.

[49] Kyu-Cheon Choi and Hyun-Jeong Lee. Fuzzy posture control for biped walking robot based on force sensor for zmp. In *SICE-ICASE, 2006. International Joint Conference*, pages 1185–1189. IEEE, 2006.

[50] Hervé Audren, Joris Vaillant, Abderrahmane Kheddar, Adrien Escande, Kenji Kaneko, and Eiichi Yoshida. Model preview control in multi-contact motion-application to a humanoid robot. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4030–4035. IEEE, 2014.

Appendix A

Preview control parameters estimation algorithm

```
function [Ke, Kx, G] = preview_control_params(T, Zh, N)
        R = 1e-6;
        Qe = 1;
        Qdpos = 0;
        Qdvel = 0;
        Qdaccel = 0;
        g = 9.8;

        % derivative state coefficient matrix
        A = [1 T T^2/2;0 1  T;0  0  1];
        B = [T^3/6;T^2/2;T];
        % coefficient matrix of state space model
        C = [1  0 -Zh/g];

        % constructing matrices for DARE
        AA = vertcat(horzcat([1], C*A),horzcat(zeros(3,1), A));
        BB = vertcat([C*B], [B]);
        RR = R;
        QQ = diag([Qe, Qdpos, Qdvel, Qdaccel]);
```

```
    % DARE equation solving
    PP = dare(AA,BB,QQ,RR);

    SS = 1.0/(RR + BB'*PP*BB);
    KK = SS*BB'*PP*AA;

    % coefficients for regulator
    Ke = KK(1,1);
    Kx = KK(1,2:4);

    Ac = AA - BB*KK;
    XX = -Ac' * PP * [1,0,0,0]';

    % first preview coefficient should be equal -Ke
    G = [-Ke];

    % compute preview coefficients
    for i=2:1:N
        G = [G SS * BB' * XX];
        XX = Ac' * XX;
    end
end
```

Appendix B

Desired ZMP generation algorithm

```
g = 9.8;
Zc = 0.85; % height of CoM
T = 1e-4; % descitisation step
N = 16000; % preview steps
sim_time = 10; % simulation time

% compute control coefficients
[Ke, Kx, G] = preview_control_params(T, Zc, N);

D_o = [1 3 5]';                     % delays
D_e = [2 4 6]';                     % delays
t = 0 : T : sim_time + N * T;  % signal evaluation time
width = 1;                            % width of each pulse

% desired ZMP signal construction
desired_zmp = 0.1 * pulstran(t, D_o, 'rectpuls', width)\
- 0.1 * pulstran(t, D_e, 'rectpuls', width);
% convert desired ZMP to simulink signal format
desired_zmp = [t' desired_zmp'];
```