Realization of Biped Walking in Unstructured Environment Using Motion Primitives

Article *in* IEEE Transactions on Robotics · December 2014

DOI: 10.1109/TRO.2014.2344431

4 authors:

Mirko Raković
University of Novi Sad

Branislav Arso Borovac
University of Novi Sad

Milutin Nikolić
University of Novi Sad

Srđan Savić
University of Novi Sad

Realization of Biped Walking in Unstructured Environment using Motion Primitives

Mirko Raković, *Member, IEEE,* Branislav Borovac, *Member, IEEE,* Milutin Nikolić, *Student Member, IEEE,* and Srđan Savić , *Student Member, IEEE*

***Abstract*—Effective and efficient motion of humanoid robots in unstructured dynamic environments is a prerequisite for their activity in the living and working environment of humans. Motion in such environments has to be adjusted all the time to suit the current conditions. This work presents a method for the synthesis and realization of the biped robot motion (walking) composed of simple movements - primitives, because any complex motion can be composed of tied primitives. The primitives are parametrized with the relationship established between the overall motion characteristics and their own parameters. In this way it is possible to achieve on-line modification at any moment. The proposed solution was tested by the simulation involving a dynamic robot model. The results demonstrate that it is possible to generate a dynamically balanced walk that can be modified on-line at any moment of its realization.**

***Index Terms*—Humanoid Robots, Humanoid and Bipedal Locomotion, Motion Primitives.**

I. Introduction

THE motion of robots acting in living and working environments of humans cannot be programmed in advance because the environment is unstructured and dynamic. This means that it is not possible to know in advance the exact arrangement and positions of the objects in the robot's immediate environment. Therefore it is necessary to ensure that the decision about motion characteristics (direction, speed, etc.), generation of the trajectory and its realization are performed on-line. Additional specificity of such an environment is that the ground floor may not always be perfectly flat and uniform (stairs, different ground levels, sills,...) and wheels are not able to ensure required mobility. Locomotion on two legs seems to be the only appropriate way. However, biped locomotion is not an easy and simple engineering task. At least two requirements have to be fulfilled simultaneously: legs have to be driven in such a way to ensure the system be "transferred" to a desired location along the selected path, while whole body motion has to be coordinated in such a synergetic manner as to permanently ensure dynamic balance.

However, looking at humans, walking itself appears to be simple and easy for realization. Neurological studies of human walk [1]–[4] have shown that certain modes of leg motion become patterns that are constantly repeated. However, all steps do not need to be identical. The shape of the movements that constitute the pattern are often changed and modified to adapt to an instantaneous situation. If the robot has a humanoid structure whose characteristics are similar to those of man, then it is possible for gait realization to apply the same concept.

This work proposes an approach to the synthesis and realization of the modifiable walk of biped humanoid robots, which is based on the generation of complex motion by composing it from a set of predefined simple basic movements, called motion primitives. Based on the information about the current state of the environment, the proposed approach enables on-line modification of the walking speed, step length, height of the leg during the swing phase and motion direction. To achieve this, the system appropriately changes the parameters of the primitives during the motion while constantly ensuring the dynamic balance.

II. Previous Work

The synthesis of the artificial gait of humanoid robots has for a long time been based on an approach that is still in use and which assumes the calculation of the reference trajectories of all joints which are afterwards realized as accurately as possible. The first method for synthesis of a dynamically balanced gait was the semi-inverse method introduced by Vukobratović [5], [6]. In this approach, for one part of the system (legs), the motion is prescribed. The motion of the rest of the system is determined as such to ensure a dynamic balance while walking, i.e. zero-moment point (ZMP) should be at a reference position inside the support area. Numerous methods that have appeared since represent only modifications and upgradings of this basic method.

With the majority of modern humanoid robots the gait is generated by prescribing the desired position of the feet on the ground [7], [8] along the path by which the robot should walk, while additionally ensuring its dynamic balance. When footprint positions are set on the ground surface and the time instants are placed for the robot to reach the given positions this then defines the direction of movement as well as speed. This data serves as the basis for the determination of the trajectories of the feet, and then using inverse kinematics, it is possible to determine the changes of the joint angles that will ensure the desired feet motion. Also, based on the preset feet positions and time instants when they are to be attained, it is necessary to assign a reference ZMP trajectory. In order to

Paper is submitted for review on June 28, 2013.

This work was supported in part by the Ministry of Education, Science and Technological Development of Republic of Serbia under Grant III44008 and in part by the Provincial Secretariat of Science and Technological Development of AP Vojvodina under Grant 2012/12345.

M. Raković, B. Borovac, M. Nikolić and S. Savić are with the Faculty of Technical Sciences, University of Novi Sad, 21000 - Novi Sad, Trg Dositeja Obradovića 6, Serbia (e-mail: rakovicm@uns.ac.rs, borovac@uns.ac.rs, milutinn@uns.ac.rs, savics@uns.ac.rs).

preserve a dynamic balance of the system, the motion of the rest of the system is calculated to follow the reference ZMP trajectory.

The method proposed in [7], allows the robot to change the subsequent position of the foot in each step. At the beginning of each half-step all necessary reference trajectories for two full steps should be generated in advance. A new position of the foot can be assigned at the moment when the leg starts the swing phase. This change brings about significant deviations of the ZMP from the reference trajectory. The deviations in the direction of motion can be reduced by adjusting the duration of the single-support phase, while the deviations in the lateral direction are compensated by applying the ZMP preview control. In [8], for a fast planning of the feet positions of the gait on a flat ground with avoiding obstacles, use is made of the off-line calculated area, in which the robot moves while checking whether some of the objects from the environment are found within. However, this approach is possible to apply only in a static scene.

In [9]–[11], the locomotion mechanism in the single-support phase was modeled using 3D Linear Inverted Pendulum Model (3D LIPM). The mass and the position of the center of mass of the 3D LIPM correspond to the mass and position of the center of mass of the overall robot, and the joint by which the inverse pendulum is fixed to the ground is the robot's ankle. In [11], the authors additionally upgraded the gait generation in real time using the 3D-LIPM model, by introducing a ZMP preview controller.

Motion of the humanoid robots can also be defined on the basis of recorded (and adapted) motion of humans. Although the term "primitives" has been used frequently for "portions of recorded motions", its notion has not been defined in a unified way by different authors. The work [12] offers a library in which each primitive represents one step. Based on the preset requirements and current state of the robot, a new step which corresponds to the requirements is selected from the library. In [13], the authors used the leg motion primitives that are obtained by segmenting the movements recorded from man (and modified so to comply with the kinematic and dynamic parameters of the robot) to preserve the form of the recorded motion while maintaining its dynamic balance.

In [14], the authors introduced the notion of Dynamic Movement Primitives (DMP), defined as the desired state of the kinematics of the extremities, obtained by prescribing in advance the values of the angles, angular velocities and angular accelerations for each robot joint. The DMP can be also described in external coordinates by assigning a desired motion of the last link in the kinematic chain. In [15], the subject of the study was the application of supervised learning of DMP, involving the learning of the movements of the hand to manipulate the objects. In the learning phase, use is made of the path integral algorithm, in which besides the target to be reached by the hand, the shape of the path that is to be performed is also taken into account. In [16], the authors described the application of the delta learning rule for joining several DMP's with the aim to obtain a precise and smooth motion so as to allow for the simulation of handwriting movements.

In [17], [18], the modular approach to movement generation is proposed based on motor primitives. The proposed model is represented as a simple trajectory generator for both discrete and rhythmic movements. The authors in [19], [20] presented a trajectories generation method for humanoid robots that is based on kinematic motion primitives derived from recorded humans locomotion trajectories. It is shown that from a small set of invariant primitives it is possible to reconstruct necessary joint trajectories to obtain modifiable gaits.

III. Motion Primitives

The notion of primitives in this paper is defined differently. Let us note that any complex movement can be decomposed into portions, such that each of them can be recognized as simple basic movements. In the case of walking, the sequence of actions is as follows: back leg deploy from ground and bend entering single support phase, transfer to front position, stretch to prepare for contact with the ground while support leg rotates at ankle to bring whole locomotion system to front position. Basic movements of which complex motion is composed (leg bending, leg stretching,...) we consider as primitives. The same primitive always involves the same joints (for leg bending there are hip, knee and ankle), but intensity of the motion at each joint can be modified any time. Fig. 1a illustrates the initial and final postures: a) for leg bending, b) for leg stretching, c) for inclining the robot forward and d) for making foot surface contact.

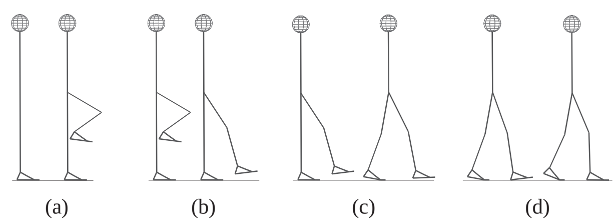


Fig. 1: Stick diagrams of the primitives: a) leg bending; b) leg stretching; c) inclining the robot forward; d) making foot surface contact.

The method we proposed in this work for the synthesis and realization of robot walk involves tied-up primitives that are parametrized and can be changed on-line. For example, in order to overcome an obstacle appearing on the robot's path, the robot has to adjust its pace, to land the foot at an exact distance from the obstacle, and then in the swing phase, bend the leg to the height needed to step over. To go around the obstacle changing the motion direction should be performed.

The characteristics of primitives can be summarized as follows:

- A primitive defines a simple movement in which more joints can be involved.
- Primitives are parametrized, so that the movement shape can be altered during its realization. The number of parameters can vary.
- Realization of a primitive does not depend on the current posture, but only on whether the robot is in such a posture from which it is possible to perform the given primitive. For leg stretching, the leg that stretches (swinging) must not be in contact with the ground; for leg bending, the

ZMP should be inside the support area of the foot which will remain in contact with the ground.

- Parameters of the primitives have to be modifiable on-line. For example, if a robot, in the course of transferring the swing leg from back to front position, faces an obstacle, it can instantaneously change the parameter that determines the stride (in the case the robot wants to stop before the obstacle), or make a sufficiently long step if the robot wants to step over it.
- Primitives can be tied together in a sequence.
- One and the same primitive can be included in the realization of different movements (leg bending is used for a walk on a flat surface and in climbing a staircase).
- Several primitives can be realized in parallel, but with different joints involved.

The approach presented differs in several important points from those mentioned earlier. Firstly, a primitive is a simple movement which is realized by simultaneous and synchronized motions of a number of joints. There are no limitations in the defining of new primitives, either in the number of parameters or in the number of joints that will be activated in its realization. Secondly, the motion is composed by combining and tying different primitives, without having a reference trajectory that was prescribed in advance. Each primitive is parametrized, so that a form of the realization is adjusted by choosing the appropriate parameters. To obtain a desired motion, it is only necessary to fix the sequence of primitives and their parameters.

IV. Synthesis of Walks and its Realization

Motion synthesis proposed in this work[1] require four steps. The first step should be performed off-line at the beginning of the motion planing phase, but all other steps are performed repeatedly at each sampling interval during motion execution.

- As a first step, the sequence of primitives needed to perform a desired motion should be defined. This is explained in detail in subsection IV-C.
- In the second step, parameters of each primitive should be defined to ensure: a) continuous and smooth movement realization and, b) its desired shape (for example, higher foot lifting to step over the obstacle).
- motion synthesized in the first two steps is based on the primitives only (see subsections IV-A1 - IV-A7), and do not take into account overall balance. However, to ensure dynamic balance of the whole system motion correction should be introduced to ensure desired position of the ZMP. How the correction is introduced to ensure dynamic balance is described in subsection IV-B. Only after that is motion considered synthesized.
- At the end of the fourth step, control voltage for each joint should be determined, since all joints are actuated by DC motors with permanent magnets.

[1]Set of primitives we use here is defined on the basis of our observation of human motion. This set is not intended to be complete for any movement. For another motion (for example, going up or down stairs) it can be changed by adding new primitives. It is interesting to note that different motions are surprisingly similar by the sequence of tied primitives and often the same primitives are included in different movements.

A. Realization of Motion Primitives

Every movement is defined on the basis of the task to be realized and modified on the basis of the current situation. If the locomotion mechanism moves and leg bending in the single-support phase performed by the swing leg has just been completed, leg stretching should smoothly continue without stopping. Trajectory of the foot, from current position to target, is not known in advance and it is generated continuously (at each sampling interval). The trajectory of the foot is determined by the trajectory of coordinate frame O_A which is attached to the heel, while the target is represented by the 6-component vector of the position and orientation of coordinate frame O_B to which coordinate frame O_A should be brought (Fig. 2). If new requirements are imposed during motion execution, the trajectory generated after that moment will smoothly change to comply to them [21].

Each primitive has an execution speed defined by the speed of frame O_A. In case of foot, this refers to the speed of transferring it from back to front position aptly named cruising speed $\mathbf{s}_c$. However, at the starting phase, a cruising speed has to be reached whereupon the primitive will end in a way appropriate for movement continuation. Cruising velocity of the O_A should additionally be modified as to ensure movement smoothness.

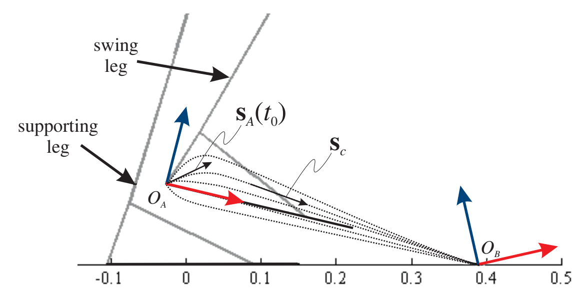


Fig. 2: Stick diagram for the robot's feet in swing phase with possible paths of O_A in dependence of the initial velocity.

The shape of the O_A trajectory to target O_B depends on the intensity and direction of the velocity $\mathbf{s}_A$ at the trajectory starting point. Let us suppose that $\mathbf{s}_A$ at the last point of leg bending (this point coincide with first point of leg stretching primitive) has direction of $\mathbf{s}_A\left(t_0\right)$ as shown in Fig. 2. The velocity $\mathbf{s}_A$ should be changed in order to reach direction and intensity of cruising speed $\mathbf{s}_c$ (Fig. 2). To be achieved smoothly, the desired velocity $\mathbf{s}_A$ is calculated as:

$$\mathbf{s}_A\left(t_i\right) = \left(1 - b\left(t_i\right)\right)\mathbf{s}_A\left(t_0\right) + b\left(t_i\right)\left[\begin{smallmatrix} v_i\cdot\mathbf{p} \\ \omega_i\cdot\mathbf{o}\end{smallmatrix}\right] \qquad (1)$$

where the $\mathbf{p}$ and $\mathbf{o}$ represent the unit vectors of the position and orientation deviation of coordinate frame O_A from O_B $(\Delta\vec{p} = \vec{p}_B - \vec{p}_A$ and $\Delta\vec{o} = \vec{o}_B - \vec{o}_A)$. In (1), the coefficient $b\left(t_i\right)$ changes linearly during the prescribed time interval from 0 to 1. We set this interval to 0.2s. This ensures a gradual change of the velocity $\mathbf{s}_A$ direction and intensity from the initial value to the value that will lead the coordinate frame O_A to the target position and orientation.

In (1), the intensities of the linear and angular instantaneous velocities v_i and ω_i depend on the cruising speeds v_c and ω_c, which are set by the primitive parameters[2]. To ensure a gradual change of the intensities of v_i and ω_i, they are calculated as:

$$v_i = \begin{cases} v_i+\Delta v & \text{if } v_i<v_c-\Delta v \\ v_i-\Delta v & \text{if } v_i>v_c+\Delta v \\ v_i & \text{otherwise} \end{cases}$$

$$\omega_i = \begin{cases} \omega_i+\Delta \omega & \text{if } \omega_i<\omega_c-\Delta \omega \\ \omega_i-\Delta \omega & \text{if } \omega_i>\omega_c+\Delta \omega \\ \omega_i & \text{otherwise} \end{cases}$$

where, Δv and $\Delta\omega$ are the empirically determined increments of the linear and angular velocities.

The single support phase has to be ended by establishing the double-support phase. After initial contact, foot has to land safely to the ground. In order to ensure gradual stopping the intensities of v_i and ω_i are to be reduced when O_A comes sufficiently close to the target. In the case of leg stretching, the velocities v_i and ω_i are proportional to $|\Delta\vec{p}|$ and $|\Delta\vec{o}|$.

Having thus determined the value $\mathbf{s}_A(t_i)$ at each time instant, and using inverse kinematics, the desired joint angular velocities for each primitive, can be calculated:

$$\dot{\mathbf{q}}^{\mathbf{p}_j} = \mathbf{J}_A^{+}(\mathbf{q}^{\mathbf{p}_j})\,\mathbf{s}_A(t_i)$$

where $\mathbf{J}_A^{+}$ is the Moore-Penrose pseudo-inverse Jacobian calculated for O_A. The vectors $\mathbf{q}^{\mathbf{p}_j}$ and $\dot{\mathbf{q}}^{\mathbf{p}_j}$ represent the angles and angular velocities of the joints involved, while the set $\mathbf{p}_j$ contains their indices. The angular velocities at the other joints do not change due to the leg's primitives, but they can change due to the simultaneous realization of some other one. On the basis of the procedure described, it is possible for the currently executing *n* primitives, to determine at any moment the angular velocities $\dot{\mathbf{q}}^{\mathbf{p}_1}$, $\dot{\mathbf{q}}^{\mathbf{p}_2}$ ... $\dot{\mathbf{q}}^{\mathbf{p}_n}$. For now, it is not possible to simultaneously realize two primitives that use the same joints but we are planning to do this in the future. Prioritization of tasks execution is one possible approach to solve this.

For the gait synthesis, we defined and used five different primitives that are realized by the legs (in both, single or double-support phase), along with one primitive that is realized by the trunk and one by the arms. The primitives that are realized by the legs when the robot is in the single-support phase are:

- bending of the swing leg (deployment of the foot from the ground, leg contraction and its transfer toward front)
- stretching of the leg in the swing phase
- inclining the robot forward.

The primitives realized during the double-support phase are:

- making the foot surface contact after the heel strike
- transferring the body weight onto the subsequent supporting leg.

The primitives that are realized by the trunk and arms are:

- maintenance of the trunk upright
- arms swinging during the walk.

For each primitive, the conditions for both the start and the end of its execution are prescribed. In the following text we will describe in detail each of the primitives and show stick diagrams illustrating the motion.

[2]Primitive parameters automatically comply with parameters of walk.

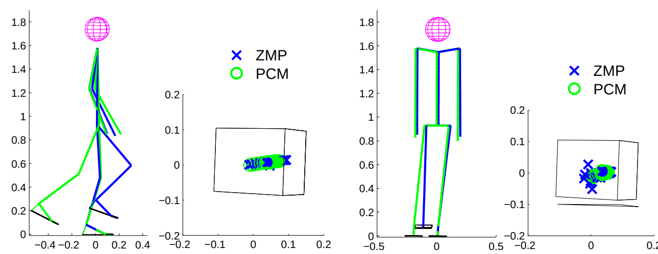
1) Leg Bending (LB): The right leg bending[3] can start if the left leg is in surface contact with the ground. To begin, the leg bending system should be in such posture as to remain balanced when back leg deploys from the ground, i.e. if the ZMP is within the safety zone[4] of the support area formed by the left foot with the ground surface.

The primitive parameters (denoted by C) for leg bending are ${}^{Speed}C_{LB}$, ${}^{Height}C_{LB}$ and ${}^{Turn}C_{LB}$. The ${}^{Speed}C_{LB}$ determines the cruising speed (both v_c and ω_c) of the coordinate frame O_A attached to the heel tip of the right foot. ${}^{Height}C_{LB}$ and ${}^{Turn}C_{LB}$ determine the height to which the foot should be lifted and the rotation of the foot is about the z axis of the coordinate frame attached to the base link (pelvis).

To calculate the desired joints angular velocities $\dot{\mathbf{q}}^{\mathbf{p}_j}$ for the leg bending, the function f_{LB} is introduced as:

$$\dot{\mathbf{q}}^{\mathbf{p}_j} = f_{LB}\left({}^{Speed}C_{LB}, {}^{Height}C_{LB}, {}^{Turn}C_{LB}, \mathbf{q}, \dot{\mathbf{q}}, LEG\right) \tag{2}$$

where $\mathbf{q}$ and $\dot{\mathbf{q}}$ are current joint angles and angular velocities whereas the parameter LEG determines whether the primitive is realized by the left or right leg. Fig. 3 shows two different cases. In Fig. 3a leg bending starts when biped is in the single support phase, while Fig. 3b illustrates the case when biped is standing still and then motion starts. The figure also depicts trajectories of the projection of center of mass (PCM) and zero moment point (ZMP) within the footprints (consisting of two trapezoidal boxes: the main base of the foot and the toe). Leg bending starts by back foot deployment from the ground that instantly and significantly reduces support area size. To preserve the robot's dynamic balance PCM and ZMP must already be within the support area beneath the left foot.



(a) (b)

Fig. 3: Stick diagram for the initial and final posture of the robot, and the trajectories of PCM and ZMP during the leg bending when the robot in the initial moment is: a) in the single-support phase; b) in the double-support phase.

2) Leg Stretching (LS): The parameters of this primitive are ${}^{Speed}C_{LS}$, ${}^{Stride}C_{LS}$, ${}^{Height}C_{LS}$, ${}^{Angle}C_{LS}$, and ${}^{Turn}C_{LS}$. The ${}^{Speed}C_{LS}$ determines both linear and angular components of the cruising speed v_c and ω_c. ${}^{Stride}C_{LS}$ determines how far the foot will stretch in front of the robot while ${}^{Height}C_{LS}$ determines the height to which the foot will be lifted from the ground. ${}^{Angle}C_{LS}$ determines the angle at which the sole of the foot will make contact with the ground while ${}^{Turn}C_{LS}$

[3]For an easier explanation we will describe the realization of the primitive for the right leg bending, while the support is on the left leg. For the bending of the left leg, the procedure is identical, but with the roles interchanged.

[4]The safety zone represents the support area that is shrunk by assigning a 10-mm margin with respect to the edge of the support area.

defines the foot orientation about the z axis with respect to the pelvis. Now, like in the case of leg bending, it is possible to introduce:

$$\dot{\mathbf{q}}^{\mathbf{p}_j} = f_{LS}\left({}^{Speed}C_{LS}, {}^{Stride}C_{LS}, {}^{Height}C_{LS}, {}^{Angle}C_{LS}, {}^{Turn}C_{LS}, \mathbf{q}, \dot{\mathbf{q}}, LEG\right) \tag{3}$$

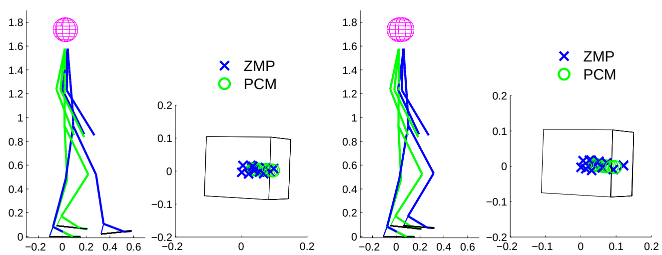
where f_{LS} is the function for the calculation of joint velocities for leg stretching, only on the basis of primitives to be realized.

To start the stretching this leg must be in the swing phase and not in contact with the ground. Fig. 4a shows the stick diagrams of the initial and final posture of the robot performing leg stretching and the trajectory of PCM and ZMP during its realization.

3) Forward Inclination (FI): It is realized by the supporting leg inclining[5] it relative to the supporting foot. The target is defined with respect to the frame attached to the heel of the supporting foot that is in contact with the ground. The coordinate frame to be brought to the target is attached to the pelvis. ${}^{Base-x}C_{FI}$, ${}^{Base-y}C_{FI}$ and ${}^{Height}C_{FI}$ determine the x, y and z coordinates of the frame O_B origin, whereas the target orientation of the pelvis is the same as of the supporting foot. ${}^{Speed}C_{FI}$ specifies v_c and ω_c, as before. The function for calculation of desired joint velocities can be written as:

$$\dot{\mathbf{q}}^{\mathbf{p}_j} = f_{FI}\left({}^{Speed}C_{FI}, {}^{Base-x}C_{FI}, {}^{Base-y}C_{FI}, {}^{Height}C_{FI}, \mathbf{q}, \dot{\mathbf{q}}, LEG\right) \tag{4}$$

where *LEG* determines which leg is to realize the primitive. The forward inclination can start if the robot is in the single-support phase.



(a) (b)

Fig. 4: Stick diagram for the initial and final posture of the robot, and the trajectories of PCM and ZMP during: a) leg stretching; b) forward inclination.

In Fig. 4b the parameter ${}^{Base-x}C_{FI}$, is adjusted so that the frame attached to the pelvis is brought above the toes link of the supporting foot. ${}^{Height}C_{FI}$ is defined so that the swinging leg, when the primitive execution is over, is still above the ground.

4) Making the Foot Surface Contact after the heel strike (FSC): This requires involvement of both legs and two frames should be brought to their target positions. The first one is attached to the pelvis which is brought to the target given with respect to the frame attached to the toes of the foot already in surface contact with the ground. The other frame is attached to the foot which is just about to make surface contact. Its target position is also given with respect to the frame of the toes. The primitive parameters are ${}^{Speed}C_{FSC}$, ${}^{Base-x}C_{FSC}$, ${}^{Base-y}C_{FSC}$ and ${}^{Height}C_{FSC}$.The cruising speeds for both coordinate frames are equal to speed defined by ${}^{Speed}C_{FSC}$.

The target of the pelvis frame is determined by x, y and z coordinate of the pelvis with respect to the toes. The target orientation of the pelvis is the same as of toes. The target orientation of the foot which rotates about the heel edge is the same as of toes of the other foot. Therefore, the function for the calculation of the joint angular velocities to realize surface contact can be written in the form:

$$\dot{\mathbf{q}}^{\mathbf{p}_j} = f_{FSC}\left({}^{Speed}C_{FSC}, {}^{Base-x}C_{FSC}, {}^{Base-y}C_{FSC}, {}^{Height}C_{FSC}, \mathbf{q}, \dot{\mathbf{q}}, LEG\right) \tag{5}$$

where, again, LEG determines the leg whose foot is about to make surface contact with the ground. The realization of the primitive is ended when the foot makes contact with the ground in at least three points.

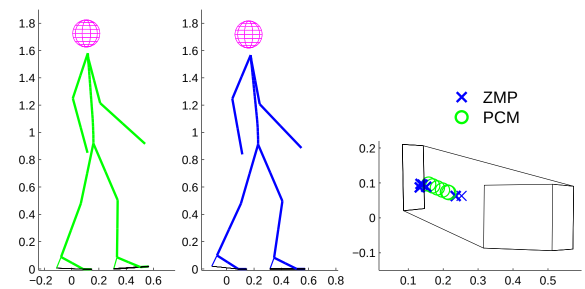


Fig. 5: Stick diagram for the initial and final posture of the robot, and the trajectories of PCM and ZMP during the foot landing to establish surface contact. The toe of the left foot is in contact with the floor at the same time as the whole surface of the right foot.

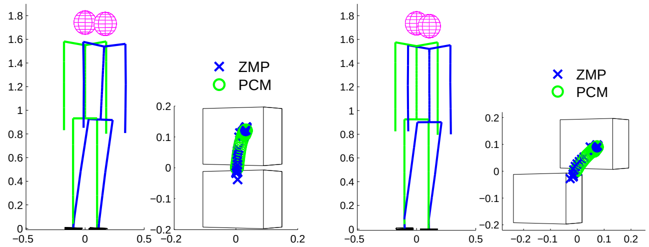
In Fig. 5 the beginning and the end of the primitive realization is shown with the trajectories of PCM and ZMP. In the beginning, the PCM and ZMP are beneath the toes of the left foot while the right foot is in contact with the ground only via the heel tip. Simultaneously with the motion of the pelvis, the right foot lowers down until it makes surface contact with the ground.

5) Transferring the Weight onto the supporting leg (TW): This primitive can be performed only in the double-support phase and involves both legs. The motion of the support leg is such to bring PCM at the desired position, while the motion of another leg is such as specified to ensure its foot remains in contact with the ground. The parameters ${}^{Speed}C_{TW}$, ${}^{Base-x}C_{TW}$, ${}^{Base-y}C_{TW}$ and ${}^{Height}C_{TW}$ denote the same as in the previous example. The function for the calculation of the joint angular velocities for this primitive is:

$$\dot{\mathbf{q}}^{\mathbf{p}_j} = f_{TW}\left({}^{Speed}C_{TW}, {}^{Base-x}C_{TW}, {}^{Base-y}C_{TW}, {}^{Height}C_{TW}, \mathbf{q}, \dot{\mathbf{q}}, LEG\right) \tag{6}$$

where, the parameter LEG determines on which leg the weight is transferred. Fig. 6 shows two examples of the same primitive execution with different initial states. In the initial moment of the first case, the robot stands still with the feet one by another. In the second case the robot steps forward by the left foot and the joint velocities at the initial moments are different from zero.

[5]This primitive inclines the entire system, but some joints can simultaneously perform another primitive (for example, keeping trunk upright). So some segments of the biped may not be inclined as we would expect.



(a) (b)

Fig. 6: Stick diagram for the initial and final posture of the robot and the trajectories of PCM and ZMP during transferring the weight onto the supporting leg in the initial posture: a) the robot stands still, the feet are one by another, b) the robot stands astride and the left foot is in front of the right one

6) Keeping the trunk Upright (UP): The trunk has a twofold role: to maintain dynamic balance and to keep its upright posture. The trunk's task to be realized is decided upon the ZMP position at the moment. If dynamic balance is jeopardized the trunk is used to prevent the system from falling down and to return the ZMP back. When dynamic balance is not jeopardized, it is possible to correct deviations of the trunk from its upright position. Transition from one to another task should be gradual.

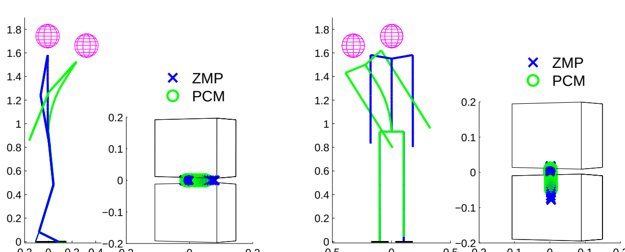
The only parameter of this primitive is ${}^{Speed}C_{UP}$, which determines the cruising speeds v_c and ω_c as:

$$v_c = K_{TR}{}^{Speed}C_{UP} \text{ and } \omega_c = K_{TR}{}^{Speed}C_{UP}$$

where K_{TR} is a variable coefficient in the range $[0\ 1]$, which changes depending on the assigned task. As described in [22] a zone has been formed around the reference position of ZMP. If ZMP is within this zone (close enough to the reference position) then dynamic balance is not jeopardized and the trunk can be used to correct its upright posture by gradually increasing the coefficient K_{TR} to 1. If ZMP is out of this zone then dynamic balance is jeopardized and the trunk is used to prevent a system overturn. In that case the value of the coefficient K_{TR} gradually decreases to 0.

The function for the calculation of the joint angular velocities for the primitive of keeping the trunk upright can be introduced in the following way:

$$\dot{\mathbf{q}}^{\mathbf{p}_j} = f_{UP}\left({}^{Speed}C_{UP}, \mathbf{q}, \dot{\mathbf{q}}\right). \tag{7}$$



(a) (b)

Fig. 7: Stick diagram for the initial and final posture of the robot and the trajectories of PCM and ZMP during keeping the 10-link trunk upright: a) in the initial posture the trunk is inclined forward; b) in the initial posture the trunk is inclined laterally

Fig. 7 shows cases when the trunk is initially inclined forward, and when the inclination is lateral. In both cases the robot returns successfully to the upright posture.

7) Arms swinging (AS): The coordinate frames are attached to the forearm tips and their target positions depend on the instantaneous position of the coordinate frame attached to the foot of the opposite leg. When the right foot is in front of the body, the left arm should also be in front of the body, and vice versa.

This primitive has only one parameter, ${}^{Speed}C_{AS}$, which determines the cruising speed v_c and ω_c for the both arms. The function for the calculation of the desired joint angular velocities for arm swinging is given by:

$$\dot{\mathbf{q}}^{\mathbf{p}_j} = f_{AS}\left({}^{Speed}C_{AS}, \mathbf{q}, \dot{\mathbf{q}}\right) \tag{8}$$

B. Preservation of Dynamic Balance and Joint Motion Control

Gait has to fulfil functional requirements (desired gait type) and simultaneously, dynamic balance has to be ensured. The balance control we applied is based on an approach presented in [22] where the zone around the desired ZMP position (at each time instant its position changes) has been predefined. If the ZMP is inside this zone then that system is dynamically balanced, i.e. it is sufficiently close to the ZMP reference position and the control system is taking care of the robot's posture. If the ZMP is out of it, focus of the control has to be changed to the task of ensuring dynamic balance. The controller permanently switches between those two tasks, not instantaneously, but gradually. There are two basic ideas behind such control. The first one is to not allow occurrence of a situation when dynamic balance of the system is seriously endangered and corrections are applied while deviations are small. The second one is that any applied correction or switching between different control strategies should be "smooth" to keep the ZMP within the support area. In such a case we can keep biped walking permanently and dynamically balanced.

Another issue that should be clarified is the relationship between PCM and ZMP. If walking speed is moderate (like in our basic motion) it can be expected that PCM is most of the time within the support area. However, the ZMP position depends on all forces acting within the system (including gravity) in which PCM and ZMP positions don't necessarily have to but could coincide. (This is particularly true in case of disturbances.) If their positions are in close proximity this reflects quality of proposed control. We should point out that even in the example of basic motion (Fig. 12) PCM is not permanently inside the support area. Just before the double support phase is established PCM is out of the support area in walking direction. We believe that ZMP and PCM positions vary with the way of walking and walking speed. Much detailed research is needed to get deeper insight.

To generate gait, a sequence of primitives must first be defined constituting gait based on overall walking characteristics (direction, walking speed,...). As illustrated in Fig. 8, the block diagram consists of three parts. The input in the first block is the desired overall motion with its parameters. In this block, on the basis of specified parameters, functions (2)-(8) for each primitive has been called and joint angular velocities $\dot{\mathbf{q}}^{\mathbf{p}}$, based solely on specified primitives are obtained. The input to this block is also the feedback of the instantaneous state of the robot (angles and angular velocities, along with the current positions of ZMP and PCM). These values are used to check the fulfilment of the conditions for starting and ending the realization of each primitive. It has to be pointed out that at this moment dynamic balance of the system has not been taken into consideration yet, as well as the desired positions of zero moment point $\mathbf{ZMP}^{\mathbf{p}}$ and projection of the center of mass $\mathbf{PCM}^{\mathbf{p}}$.

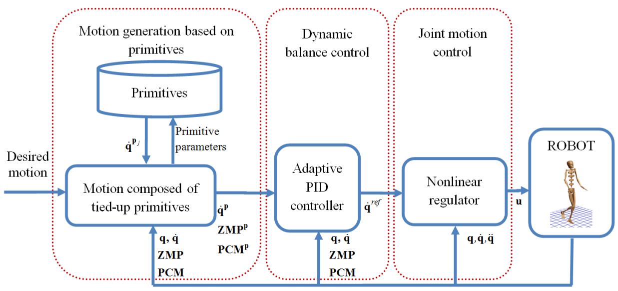


Fig. 8: Block diagram of the robot control for the realization of the motion synthesized using primitives.

The role of the second block is to ensure the maintenance of dynamic balance by modifying primitive-based angular velocities $\dot{\mathbf{q}}^{\mathbf{p}}$ defined in the first block. After modification, joint velocities are given as

$$\dot{\mathbf{q}}^{ref}(t_i) = \dot{\mathbf{q}}^{\mathbf{p}}(t_i) + \Delta\dot{\mathbf{q}}(t_i).$$

Corrections are determined on the basis of the desired and current values of ZMP and PCM, as well as on the current values of the joint angles and angular velocities $\mathbf{q}$ and $\dot{\mathbf{q}}$ by applying an adaptive PID regulator whose detailed structure is given in Fig. 9. The correction $\Delta\dot{\mathbf{q}}$ is calculated as:

$$\begin{aligned}\Delta\dot{\mathbf{q}}(t_i) = {} & \mathbf{a}_{ZMP}\Big(\mathbf{K}_P^{ZMP}\Delta\mathbf{ZMP}(t_i) + \\ & + \mathbf{K}_I^{ZMP}\sum_{k=0}^{i}\Delta\mathbf{ZMP}(t_k) + \mathbf{K}_D^{ZMP}\Delta\dot{\mathbf{ZMP}}(t_i)\Big) \\ & + \mathbf{a}_{PCM}\Big(\mathbf{K}_P^{PCM}\Delta\mathbf{PCM}(t_i) + \\ & + \mathbf{K}_I^{PCM}\sum_{k=0}^{i}\Delta\mathbf{PCM}(t_k) + \mathbf{K}_D^{PCM}\Delta\dot{\mathbf{PCM}}(t_i)\Big)\end{aligned} \tag{9}$$

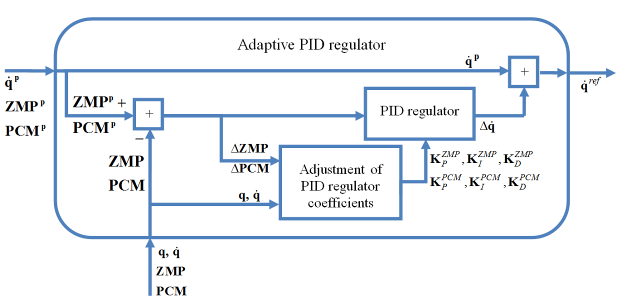


Fig. 9: Block diagram for the correction of desired joint angular velocities for maintaining dynamic balance.

The correction calculated in (9) depends on the deviations of ZMP and PCM from the desired positions. It was adopted that ZMP follow desired PCM position, i.e. the desired position of ZMP is equal to the current position of the $\mathbf{PCM}(t_i)$. This means that the deviation of ZMP is calculated from:

$$\Delta\mathbf{ZMP}(t_i) = \mathbf{PCM}(t_i) - \mathbf{ZMP}(t_i).$$

The role of PCM is also delicate. During locomotion, abrupt change in PCM position should be avoided. Hence a target position of PCM ($\mathbf{PCM}^{target}$) was introduced. At each sampling interval, $\mathbf{PCM}^{\mathbf{p}}$ moves toward $\mathbf{PCM}^{target}$ gradually and proportionally to the walking speed. Thus, the deviation of PCM ($\Delta\mathbf{PCM}$) from the desired position is calculated from:

$$\Delta\mathbf{PCM}(t_i) = \mathbf{PCM}^{\mathbf{p}}(t_i) - \mathbf{PCM}(t_i).$$

The $\mathbf{PCM}^{target}$ is not fixed but depends on the posture to which the robot is to be brought. For example, for forward inclination, the $\mathbf{PCM}^{target}$ should be beneath the foot link, or, if the robot needs to support itself on the toes, the $\mathbf{PCM}^{target}$ should be beneath the link of toes. In (9) the $\mathbf{K}_P^{ZMP}$,$\mathbf{K}_I^{ZMP}$,$\mathbf{K}_D^{ZMP}$,$\mathbf{K}_P^{PCM}$,$\mathbf{K}_I^{PCM}$ and $\mathbf{K}_D^{PCM}$ represent the variable coefficient matrices of proportional, integral and differential action of the PID regulator. The block for the adjustment of the regulator coefficients determines which joints are to be engaged in the action of correction and the way this will be done. The coefficients are adjusted in the following way. First, it is checked whether the feet of both legs make surface contact with the ground. If one of the feet has no surface contact, then the coefficients for that leg are equal to 0. If the foot makes surface contact, then it is necessary to change the coefficients dependent upon the instantaneous value of $\Delta\mathbf{ZMP}$ and $\Delta\mathbf{PCM}$. The coefficients in each iteration are increased/decreased by 1% of its maximum value. If the $\Delta\mathbf{ZMP}$ is greater than the predefined threshold (in this case the threshold is set to 3 cm), then the coefficients $\mathbf{K}_P^{ZMP}$,$\mathbf{K}_I^{ZMP}$ and $\mathbf{K}_D^{ZMP}$ increase[6]. When the value of the $\Delta\mathbf{ZMP}$ is less than the threshold (at that moment, dynamic balance is not directly jeopardized), the coefficients $\mathbf{K}_P^{ZMP}$,$\mathbf{K}_I^{ZMP}$ and $\mathbf{K}_D^{ZMP}$ gradually decrease to 0. The same rule is also valid for the $\mathbf{PCM}$ position ($\mathbf{K}_P^{PCM}$,$\mathbf{K}_I^{PCM}$ and $\mathbf{K}_D^{PCM}$). If the $\Delta\mathbf{PCM}$ is greater than the threshold, then the coefficients $\mathbf{K}_P^{PCM}$, $\mathbf{K}_I^{PCM}$ and $\mathbf{K}_D^{PCM}$ increase to their maximum values, and when the $\Delta\mathbf{PCM}$ is less than the threshold, the coefficients gradually decrease to 0. The coefficients $\mathbf{K}_P^{ZMP}$,$\mathbf{K}_I^{ZMP}$,$\mathbf{K}_D^{ZMP}$,$\mathbf{K}_P^{PCM}$,$\mathbf{K}_I^{PCM}$ and $\mathbf{K}_D^{PCM}$ for the trunk joints change in the same way as the coefficients for the legs joints.

The experiments presented in [23], [24] showed that the movements of the ankle joint in one direction and of the hip in the other direction ensured a very efficient control of the ZMP position avoiding large movement of PCM, i.e. large internal forces. In (9), the coefficients in the matrices $\mathbf{a}_{ZMP}$ and $\mathbf{a}_{PCM}$ have the only role of determining the direction in which compensation should be performed, and they can have values either -1 or 1. At Table I are given the values of

[6] This way ensures that the maintenance of dynamic balance has a higher priority.

the coefficients $\mathbf{a}_{ZMP}$ and $\mathbf{a}_{PCM}$ for all the joints at which corrections are performed.

TABLE I: The signs of the coefficients $\mathbf{a}_{ZMP}$ and $\mathbf{a}_{PCM}$ needed to define direction of corresponding joint rotation to ensure desired synergy

Joint	$\mathbf{a}_{ZMP}$	$\mathbf{a}_{PCM}$
$\vec{e}_8$ - right hip roll	+1	-1
$\vec{e}_9$ - right hip pitch	+1	-1
$\vec{e}_{11}$ - right ankle roll	-1	+1
$\vec{e}_{13}$ - right ankle pitch	-1	+1
$\vec{e}_{16}$ - left hip roll	+1	-1
$\vec{e}_{17}$ - left hip pitch	+1	-1
$\vec{e}_{19}$ - left ankle roll	-1	+1
$\vec{e}_{21}$ - left ankle pitch	-1	+1
$\vec{e}_{23}$,$\vec{e}_{25}$,...$\vec{e}_{41}$ - all trunk pitch	+1	-1
$\vec{e}_{24}$,$\vec{e}_{26}$,...$\vec{e}_{42}$ - all trunk roll	+1	-1

Fig. 10 shows two cases of the execution of the primitive for keeping the trunk upright when maintenance of dynamic balance is included and excluded. When the correction is excluded (left picture), the ZMP trajectory oscillates significantly and comes close to the edge of the support area. This means the robot barely avoided falling. In Fig. 10b, for the same speed of execution of the primitive, the maintenance of dynamic balance was included. It can be clearly seen that the ZMP does not come close to the edge of the support area and dynamic balance has been preserved.

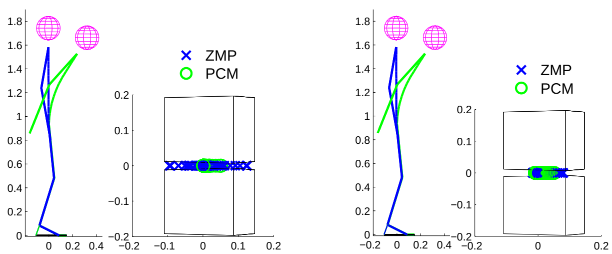


Fig. 10: Stick diagram for the robot and the trajectories of PCM and ZMP during the realization of the primitive for keeping the trunk upright, for the cases when the corrections of desired joint angular velocities for maintaining dynamic balance are: a) excluded, and b) included

By introducing corrections to preserve dynamic balance, angular velocities to be executed by joints (the reference joint velocities $\dot{\mathbf{q}}^{ref}(t_i)$) are defined. Further on, it is necessary to calculate the voltages of the motors to actuate the joints. For the realization of the reference motion at the joints we used a non-linear regulator which is a combination of feedback linearisation, sliding mode control and disturbance estimator. Since the humanoid robot, is a multivariable, coupled, and highly non-linear system, use was made of feedback linearisation in order to cancel the non-linearities in the system. The robot dynamics, including also the actuators dynamics, is presented in the state space in a controllability canonical form [25]:

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \end{bmatrix}, \ \mathbf{y} = \mathbf{x} \\ \mathbf{x} &= \begin{bmatrix} \mathbf{q}^T \ \dot{\mathbf{q}}^T \end{bmatrix}^T, \ \dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}}^T \ \ddot{\mathbf{q}}^T \end{bmatrix}^T \end{aligned} \tag{10}$$

where $\mathbf{x}$ is the system state vector, $\mathbf{y}$ is the output vector (angular positions), $\mathbf{q}$ is the vector of generalized coordinates, $\mathbf{u}$ is the vector of motor voltages in each joint, whereas $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are nonlinear state functions. The matrices $\mathbf{f}$ and $\mathbf{g}$, as well as detailed derivation of the control algorithm are given in Appendix B. By choosing the control $\mathbf{u}$ in the form:

$$\mathbf{u} = \mathbf{g}(\mathbf{x})^{-1}(\mathbf{v} - \mathbf{f}(\mathbf{x}))$$

and introducing it into (10), the system nonlinearities are cancelled and the model in the state space reduces to the system of double integrators:

$$\ddot{\mathbf{q}} = \mathbf{v}.$$

In order to apply the feedback linearization that could fully cancel the nonlinearities, it is necessary that the system model be known and utmost accurate. In reality, a model is always different from the real system. Because of the existence of inaccuracies of the model parameters and presence of external disturbances acting on the system the robust sliding mode control is used [26]. The new control $\mathbf{v}$ is selected in the following form:

$$\mathbf{v} = \ddot{\mathbf{q}}^{ref} - \mathbf{K}_D\Delta\dot{\mathbf{q}} - \mathbf{K}_P\Delta\mathbf{q} - \mathbf{K}_S\boldsymbol{\rho}(\gamma) - \hat{\mathbf{d}} \tag{11}$$

where $\mathbf{K}_D$ and $\mathbf{K}_P$ are the matrices that contain the gains of differential and proportional term of the regulator; $\mathbf{K}_S$ is the gain matrix of the sliding mode term; $\boldsymbol{\rho}(\gamma)$ is the proportional-integral function by which is approximated the non-smooth sign function and eliminated the chattering effect [27]; $\ddot{\mathbf{q}}^{ref}$ is the vector of the reference joint angular accelerations obtained by differentiating the $\dot{\mathbf{q}}^{ref}$; $\Delta\mathbf{q} = \mathbf{q} - \mathbf{q}^{ref}$ is the vector of the error of the angular positions, and $\hat{\mathbf{d}}$ is the vector of estimated disturbances at the joints.

C. Synthesis of on-line modifiable walk

Walking with moderate speed on a flat level ground surface is what we consider a basic motion that can be modified (speed up or down, increase or decrease stride, etc.) by simply changing overall requirements. As parameters describing overall motion we introduce walk speed W_{Speed}, height to which the foot is lifted during the swing phase W_{Height}, step length W_{Lenght}, and walking direction W_{Dir}. For basic motion these parameters have following values: $W_{Speed} = 1$, $W_{Height} = 1$, $W_{Lenght} = 1$, $W_{Dir} = 0$ (no changing direction).

If value of these overall parameters are changed, parameters of the primitives should be changed immediately and accordingly to comply with new requirements. Thus, the walk speed (W_{Speed}) has a direct influence on execution speed of each primitive (i.e. on cruising speeds v_c and ω_c). The height to which the foot is to be lifted (W_{Height}) influences the parameters for leg bending, the step length (W_{Lenght}) influences the parameters for leg stretching, and the desired walk direction (W_{Dir}), which is given as the turn angle with respect to the x axis of the fixed coordinate frame, influences the parameters of the primitives for leg bending and leg stretching. Of course, command for change of any of those four commands can be issued any time, but the effect will be visible when a corresponding primitive has been performed. For example, if increase of walking speed was required, cruising speeds of all performing primitives will start to increase immediately, but if increase of height to which foot is to be lifted has been required it will become effective when system approach obstacle to step over.

A half-step has been divided into phases according to a sequence of primitives constituting walk that is realized by legs. In each phase target position of the PCM, i.e. $\mathbf{PCM}^{target}$, has been defined for the primitives that are executing. Simultaneously with the primitives that are realized by the legs, irrespective of the current phase, the primitive for keeping the trunk upright and the primitive for arm swinging are constantly executed during the walk.

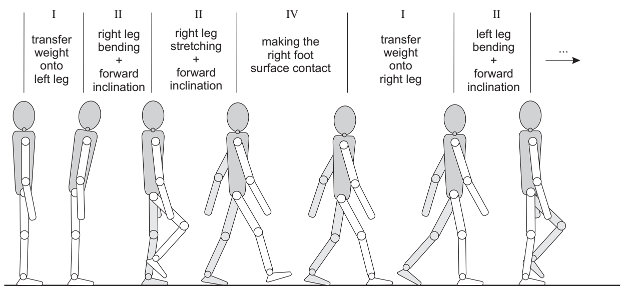


Fig. 11: Decomposition of the walk into the phases that can be realized by introduced primitives.

Let us suppose that the motion begins from the posture in which the robot is standing still in the double-support phase (Fig. 11). During the first phase, the robot transfers its weight onto the left (supporting) leg. This action is taking place either if robot is standing still and motion has just started or system is already in motion, as well as in the double support phase when load has to be transferred to another leg. The primitive parameter ${}^{Speed}C_{TW}$ is proportional to the desired walk speed W_{Speed} in a following way:

$$^{Speed}C_{TW} = W_{Speed} \cdot {}^{Speed}C_{TW_1}$$

where ${}^{Speed}C_{TW_1}$ is the speed of transferring weight onto the supporting leg for basic motion. Subscript 1 denotes that transferring motion takes place during phase 1. The position of $\mathbf{PCM}^{target}$ is set beneath the supporting foot (the left one) in the single-support phase. The parameters ${}^{Base-x}C_{TW}$, and ${}^{Base-y}C_{TW}$ have the same values as $\mathbf{PCM^p}$, which means that in the course of execution of this primitive, the base coordinate frame is brought to the position above the left foot.

During the second phase (subscript 2), the right leg deploys from the ground and the whole system inclines forward. Because of that, the primitives for leg bending and forward inclination are realized in parallel. All parameters of the primitive for leg bending depend on the gait parameters in the following way:

$$^{Speed}C_{LB} = W_{Speed} \cdot {}^{Speed}C_{LB_2},$$

$$^{Height}C_{LB} = W_{Height} \cdot {}^{Height}C_{LB_2} \text{ and } {}^{Turn}C_{LB} = W_{Dir}$$

where ${}^{Speed}C_{LB_2}$ and ${}^{Height}C_{LB_2}$ are the speed of leg bending and height to which the foot is lifted when the basic motion is performed. The parameter of the speed for forward inclination ${}^{Speed}C_{FI}$ is proportional to the walking speed, so that:

$$^{Speed}C_{FI} = W_{Speed} \cdot {}^{Speed}C_{FI_2},$$

where ${}^{Speed}C_{FI_2}$ is the speed of forward inclination for basic motion. The position of $\mathbf{PCM}^{target}$ during the second phase is at the midpoint between the center of mass of the foot and the center of mass of the link of toes. The parameters ${}^{Base-x}C_{FI}$ and ${}^{Base-y}C_{FI}$ have the same values as the position of $\mathbf{PCM^p}$.

In the third phase, the swing leg stretches, and it is brought to the front of the body, while inclining simultaneously forward, in order for the system be prepared to pass to the forthcoming double-support phase. The primitives for leg stretching and forward inclination are executed in parallel. During the third phase, the position of $\mathbf{PCM}^{target}$ is beneath the link of toes. The speed of the forward inclination is still proportional to the walk speed, and it is calculated by the following equation:

$$^{Speed}C_{FI} = W_{Speed} \cdot {}^{Speed}C_{FI_3},$$

where ${}^{Speed}C_{FI_3}$ is the speed of forward inclination during the third phase for the basic motion. The parameters for leg stretching that depend on the gait parameters are calculated in the following way:

$$^{Speed}C_{LS} = W_{Speed} \cdot {}^{Speed}C_{LS_3},$$

$$^{Stride}C_{LS} = W_{Lenght} \cdot {}^{Stride}C_{LS_3},$$

$$^{Height}C_{LS} = W_{Height} \cdot {}^{Height}C_{LS_3} \text{ and } {}^{Turn}C_{LS} = W_{Dir},$$

where ${}^{Speed}C_{LS_3}$,${}^{Stride}C_{LS_3}$ and ${}^{Height}C_{LS_3}$ are the parameters of the speed and the target position of the foot in the third phase for the basic motion. The parameters ${}^{Base-x}C_{FI}$ and ${}^{Base-y}C_{FI}$ are equal to $\mathbf{PCM^p}$.

In the fourth phase, the double-support phase begins by the foot landing on the ground to make a surface contact and the primitive for ensuring foot surface contact is executed. The primitive parameter that depends on the gait parameters is ${}^{Speed}C_{FSC}$, and it is proportional to the desired walk speed W_{Speed}. The parameter ${}^{Speed}C_{FSC}$ is calculated by the formula:

$$^{Speed}C_{FSC} = W_{Speed} \cdot {}^{Speed}C_{FSC_4}$$

where ${}^{Speed}C_{FSC_4}$ is the parameter of the primitive in the fourth phase for basic motion. The $\mathbf{PCM}^{target}$ is at the midpoint between the center of mass of the link of toes and the center of mass of the foot of the other leg. The parameters ${}^{Base-x}C_{FSC}$ and ${}^{Base-y}C_{FSC}$ coincide with $\mathbf{PCM^p}$. The ending of the fourth phase is followed by the phases which are identical to the first four. The only difference is that the primitives are now realized by the other leg. These phases are repeated cyclically till it is necessary for the robot to move. Fig. 12 shows the stick diagrams of the robot, disposition of the feet, and positions of the ZMP and PCM during the realization of the basic motion synthesized by the previously described procedure.

To make the diagram more legible a posture is shown for every 1000 iterations, and each subsequent posture is shown with gray and black stick diagram. In the course of the basic motion, the robot performed walk with average speed of 0.54 km/h. The trajectories of ZMP and PCM were at all times

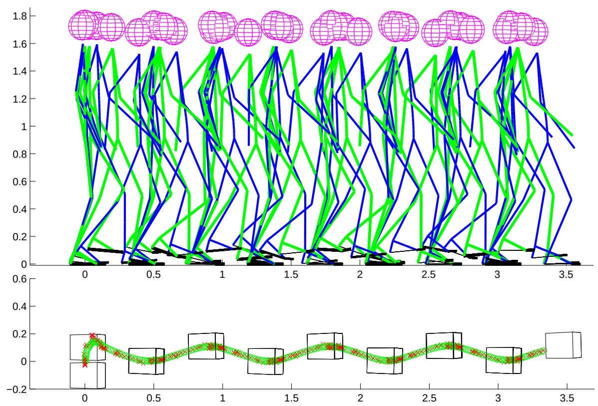


Fig. 12: Stick diagram for the robot, positions of the feet, and the trajectories of ZMP and PCM during the gait generated by using primitives.

inside the support area, i.e. the robot's dynamic balance was constantly preserved.

V. On-line Modification of the Synthesized Walk

In the previous section, walk was defined by four parameters (W_{Speed},W_{Height},W_{Lenght} and W_{Dir}). Now we will show how the change of any of those parameters influence on-line adaptation of the primitive's parameters.

Let us first discuss the example of increasing walking speed. To achieve this, it is necessary to establish the relationship between the W_{Speed} and the $\mathbf{PCM}^{target}$. If $W_{Speed} < 1$, then the $\mathbf{PCM}^{target}$ is at the spot described in Section IV-C. When a higher walk speed is required, the position of $\mathbf{PCM}^{target}$ shifts toward the interior of the foot. The larger increase in the walk speed causes a larger shift of the $\mathbf{PCM}^{target}$ (Fig. 13). This is needed because faster walking speed implies larger deviation of the ZMP from the PCM. By moving the target PCM to interior of the foot, the larger deviation to the outer part of the foot is permissible while still preserving dynamic balance, i.e. keeping the ZMP inside support area. In this case, the shift of the $\mathbf{PCM}^{target}$ with respect to its position for the basic motion is, in this particular case, at most 5 cm. If the walk speed increases from $W_{Speed} = 1$ to $W_{Speed} = 2$, the $\mathbf{PCM}^{target}$ position shifts linearly in the range of 0-5 cm. When W_{Speed} exceeds 2, then the $\mathbf{PCM}^{target}$ shift remains same (that is 5 cm). Adjustment of the position of $\mathbf{PCM}^{target}$ in the phase 3, when the robot inclines forward on the supporting leg above the link of toes, is illustrated in Fig. 13.

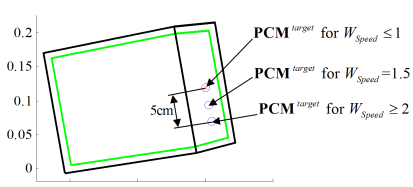


Fig. 13: Positions of $\mathbf{PCM}^{target}$ for different walk speeds during forward inclination in which the robot inclines above the link of toes of the left foot.

Several simulation examples of on-line modification of walk parameters were performed. Fig. 14 shows the walk during which the walking speed changes on-line. In the first step $W_{Speed} = 1$, in the second $W_{Speed} = 0.5$, in the third it is again $W_{Speed} = 1$, and during the fourth step $W_{Speed} = 1.5$. The other parameters of the walk do not change. Fig. 15 depicts the trajectory of the ZMP along the x and y axes as well as the time needed for the robot to realize one step at different speeds. The time to complete the second step is longer which means that this step is slower, whereas the time to complete the fourth step is shorter which means that the step is faster.

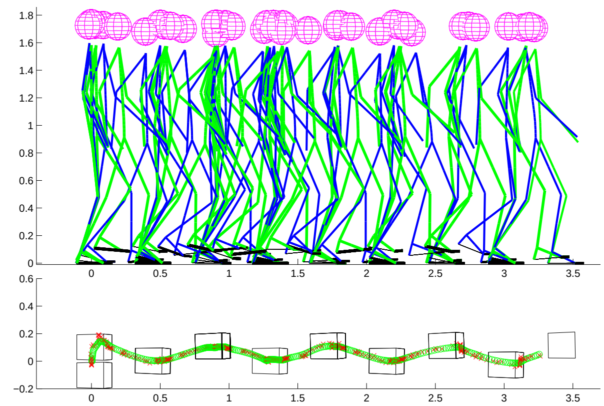


Fig. 14: Stick diagram for the robot, positions of the feet, and the trajectories of ZMP and PCM during the on-line change of the walking speed.

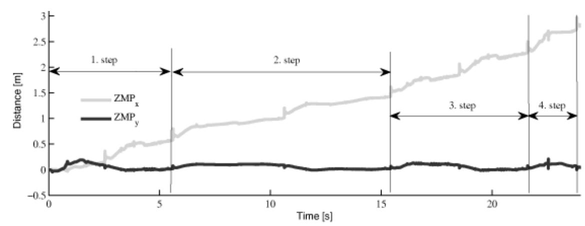


Fig. 15: Trajectories of ZMP along the x (grey line) and y (black line) axis during the on-line change of the walking speed.

The second example illustrates the changes of the step length (Fig. 16). During the first step $W_{Lenght} = 1$, during the second $W_{Lenght} = 1.25$, during the third again $W_{Lenght} = 1$, whereas during the fourth step $W_{Lenght} = 0.5$. The other parameters of the walk did not change. On the basis of the positions of the feet shown in Fig. 16 it can be seen that the second step was the longest, whereas the fourth step was the shortest.

The third example illustrates the change in the height of the foot during the swing phase (Fig. 17). During the first step, this height is $W_{Height} = 1$, during the second step $W_{Height} = 0.5$, during the third step $W_{Height} = 1.5$, whereas from the fourth step on $W_{Height} = 1$. The other parameters did not change. As can be seen from Fig. 17, the position to which the foot is lifted is the lowest during the second step, and highest during the third step.

The fourth example illustrates the on-line changes of the walking direction (Fig. 18). After each half-step, the biped turned right by $10°$. After the fourth half-step, the biped starts to turn left, and the direction in the next six half-steps changes each time by $10°$. The other walking parameters did

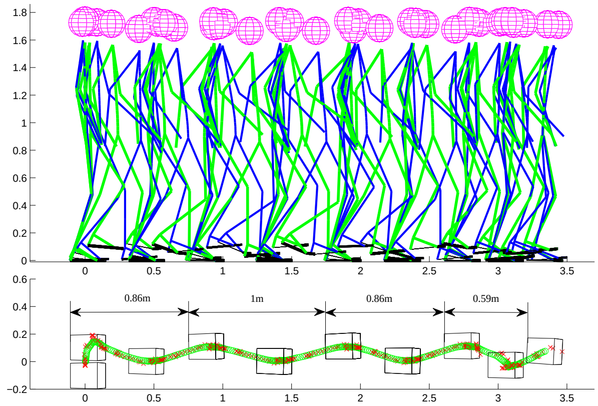


Fig. 16: Stick diagram for the robot, positions of the feet, and the trajectories of ZMP and PCM in the case of on-line change of the step length.

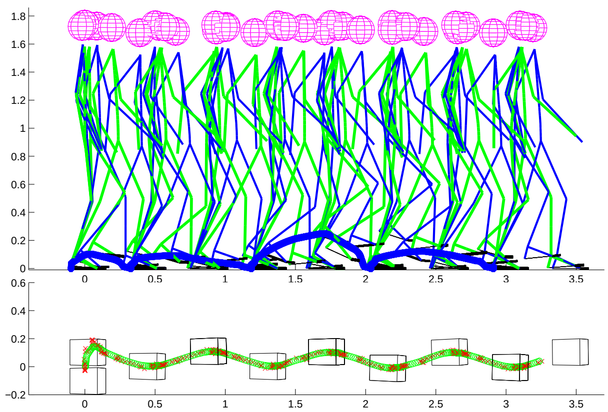


Fig. 17: Stick diagram for the robot, positions of the feet, and the trajectories of ZMP and PCM for the case of on-line change of the foot height during the walk.

not change. From the diagrams shown in Fig. 18 it can be seen that the robot changes direction of its motion by turning right for 40°, and then to the left for 60°, to walk further in the direction of 20° with respect to the initial direction.

In the course of walking, man usually changes several parameters of the walk simultaneously. When he makes shorter steps, he slows the walk down, and when he walks faster, the step length is longer. Also, when he wants to pass over an obstacle, man approaches the obstacle while adjusting on-line his step length, lifts the foot to the necessary height, and then steps over the obstacle. Fig. 19 presents the case when the robot adjusts the step length so that it can appropriately approach the obstacle to be stepped over. The step length immediately before the obstacle is $W_{Lenght} = 0.5$. When the robot approaches the obstacle, it does not stop in front of it but continues the walk with $W_{Height} = 1.5$, and $W_{Lenght} = 1.25$. After stepping successfully over the obstacle, the parameters for step length and height to which the foot is to be lifted are set to $W_{Height} = 1$, and $W_{Lenght} = 1$.

Fig. 20 presents the case when the robot is adjusting the pace, direction of the walk and height of the foot during the swing phase in order to pass between the tables and step over the bar on the ground. During the first four half-steps, the biped turns right by 5° each time. Then, in the next four half-steps it

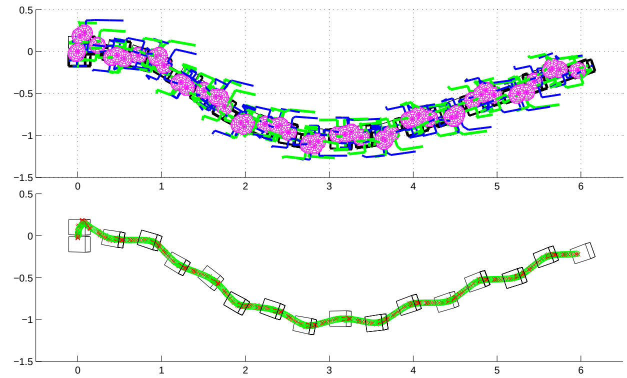


Fig. 18: Stick diagram for the robot, positions of the feet, and the trajectories of ZMP and PCM for the case of on-line change of the walking direction.

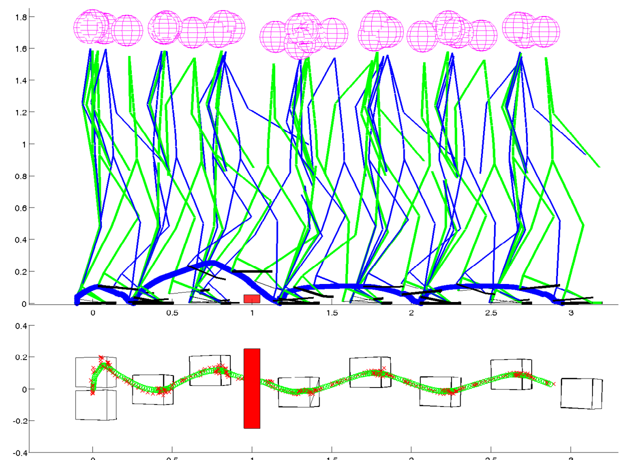


Fig. 19: Stick diagram for the robot, positions of the feet, and the trajectories of ZMP and PCM for the case of on-line change of the step length, and the foot height needed for the robot to step over the obstacle in front of it.

turns left by 5°. So, the walking direction at the end is the same as in the beginning. During all four steps, the walking speed is $W_{Speed} = 0.8$. After returning to the original direction, the walking speed is $W_{Speed} = 1$. Also, during the second step the $W_{Height} = 1.25$, i.e. the foot is lifted higher and the robot successfully steps over the small bar on the ground.

In all the examples we have shown the cases when the walking parameters change at the end of a step. This was done with the purpose to make the effects of this change more easily visible, and to enable a comparison of the steps that are performed for the different walking parameters. However, the walking parameters can be changed at any time to any value from the permissible range.

VI. Applicability of the Method

In spite of the fact that we did not have the possibility of testing this approach on the real robot and testing was done just by simulation we strongly believe this method is fully applicable to the real system. Here are some points we want to emphasize:

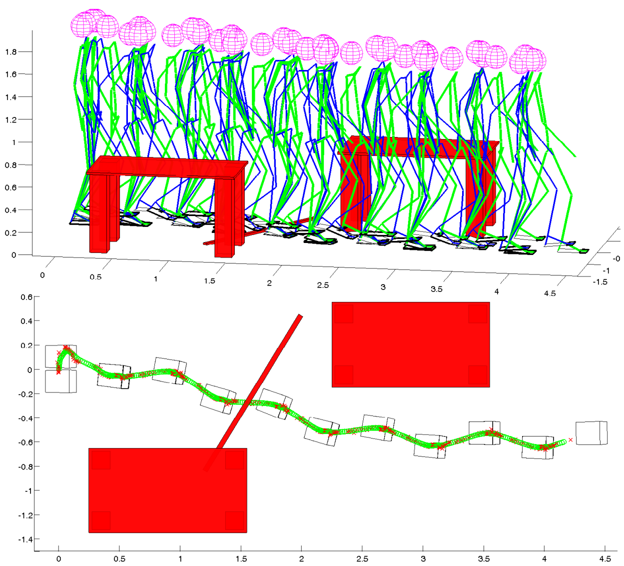


Fig. 20: Stick diagram for the robot, positions of the feet, and the trajectories of ZMP and PCM for the case of on-line change of the direction and speed of the walk to pass around the obstacle that is in front of the robot.

- In the paper we focused on showing that gait of the desired characteristics can be synthesized from sequences of modifiable motion primitives. Of course, balance control must be ensured permanently, and in approach we propose it is embedded in the synthesis procedure. However, balance control was not explained in much detail because we used our previously developed method, elaborated in [22].
- Range of admissible variation of parameters depends on kinematic and dynamic characteristics of the biped, as well as of applied actuators, and should be specified for every particular locomotion system. Parameters we used were adjusted to the biped we simulated.
- Change of walking parameters can be imposed at any moment. Changes will become visible (corresponding parameters of primitives will be changed) when primitive executions start. In case change of walking parameter was imposed while corresponding primitive was executed, change will start immediately, but gradually. Decisions about timing, i.e. when requirement for certain changes will be issued, have to be made by higher levels of cognitive system and this requires anticipation of future actions to be performed.
- Sequence of primitives (executing by same limb joints, like leg bending followed by leg stretching) should be defined in advance. But, the time instant when execution of a particular primitive will start depends on conditions that have to be fulfilled. Those conditions are incorporated in the primitive itself. For example, leg bending cannot begin before ZMP is within the safety zone of the foot supporting area of the other leg which will become the only supporting leg after foot deploys from the ground.
- Execution of primitives which employ joints not involved in currently executing primitives can start immediately, like changing walking direction.
- Inserting or removing a primitive from the sequence of already tied primitives is not possible at the moment.

The main topic of this paper is a novel approach for gait synthesis based on motion primitives, which was discussed in detail. This approach includes compensation of small disturbances because they are permanently present and cannot be avoided. However, we did not discuss compensation of large disturbances (like pushing, sudden stopping, ...) but we believe these cases can also be included into a framework similar to the one described in [23].

VII. Conclusion and Future Work

In the frame of this work we examined the problem of the synthesis and realization of bipedal walk of humanoid robots in unstructured environments. Bearing in mind that the human environment is dynamic and highly unstructured, it is necessary to ensure on-line modification of the robot motion if it is to move efficiently. In view of this, the present study proposes a new method for the realization of robots walking using primitives, which complies with these requirements. In the first place, the motion was composed by tying primitives and not taking into account the preservation of dynamic balance of the system, just the shape of motion. After that, an adaptive regulator was designed to slightly modify motion to prevent system overturn by monitoring the deviation of the ZMP from its desired position and ensuring that the motion would be changed accordingly.

Walk combined from primitives can be easily modified just by establishing the relationship between the primitive parameters and the preset overall parameters of the walk. This made it possible to automatically change the parameters of the primitives, by imposing the different walking parameters. The approach was illustrated by several examples, first, by the examples in which only one gait parameter was changed, and after that by examples in which several parameters changed simultaneously.

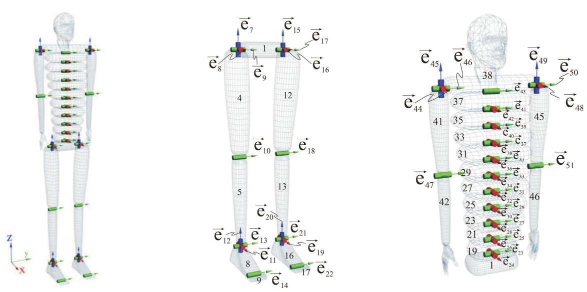
The parameters of the primitives for the basic motion were determined empirically. Range of admissible variation of parameters depend on characteristics of each particular biped. If primitive parameters are better defined resulting motion will be more anthropomorphic. This is one of our goals for further works.

This approach enables us to generate motion on-line according to the current situation in the environment. If the basic set of tied primitives is specified in advance (for example, walk on flat surface, going up and down stairs, ...) the sensory system (vision) just has to ensure information about the situation in the environment (direction of intended motion, is there an obstacle, its size and how far is it, ...) to enable the system to select appropriate primitives' parameters and walking can be performed.

Our future work will also be directed toward the introduction of new primitives for the realization of the tasks such as object manipulation during the walk (object grasping, object manipulation, etc.). Also, the attention will be focused on the synthesis of the movements for ascending and descending the stairs. It will be endeavoured that the transition from one synthesized movement to the subsequent one goes smoothly and without unnecessary stopping. It should also be mentioned that within our scope of interest is to apply this method of walking on irregular terrains such as ruins and debris after catastrophic events.

Appendix A
Dynamic Model of Biped Robot

The robot model consists of several kinematic chains whose links are interconnected with rotational joints with one degree of freedom (DOF) [28]. Fig. 21 shows the kinematic structure with 46 links, which is used for the simulations presented in this work. The joints with more DOFs (hip, ankle, shoulder, and trunk) were modeled as a set of more one-DOF joints connected with massless links of infinitesimally small lengths (fictitious links). For example, the spherical joint at the right hip was modeled by three one-DOF links. The unit vectors of the axes about which rotations are performed (see 21b) denoted as $\vec{e}_7$, $\vec{e}_8$ and $\vec{e}_9$) are mutually orthogonal and connected to the fictitious links 2 and 3, which, for the sake of simplicity, were not presented in Fig. 21. The foot was modeled as a two-link structure with the soles in the form of trapezes. The contact between the foot and the ground is determined by six characteristic points. The four contact points are at the corners of the foot body sole, whereas the other two points are at the corners of the link of toes. With the aid of these six points it is possible to describe all essential configurations of the foot-ground contact. If the number of points that are in contact with the ground is equal to or higher than three, then there will exist a surface contact between the foot and the ground. Since there are two feet, the overall number of points that are to be taken into account is 12. Therefore, for the existence of the support area it is necessary that at least three non-collinear points (on each foot in the single-support phase, or feet in the double-support phase), of the overall 12 points that are available, be in contact with the ground.



(a) (b) (c)

Fig. 21: Kinematic structure of the robot model: a) the robot as a whole; b) the legs; c) the trunk and the arms.

In order to include the effects arising at the moment of establishing/breaking the foot-ground contact, the foot was modeled as a rigid body with a viscoelastic layer on the sole, which was modeled as an isotropic Kelvin-Voigt material [29], [30], [30]. The time instant of establishing/breaking contact is calculated by using slack variables [32].

The model of the overall system is given by following equation:

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{h}_0 = \boldsymbol{\tau} + \sum_{i \in S} \mathbf{J}_i^T \begin{bmatrix} \mathbf{F}_i \\ \boldsymbol{\delta}_i \times \mathbf{F}_i \end{bmatrix} \tag{12}$$

where S stands for the indices of the points on the feet that are in contact with the ground, and $\mathbf{F}_i$ and $\boldsymbol{\delta}_i$ represent the force that appears at the i-th contact point and the deformation of the viscoelastic layer. The actuators at the powered DOFs were modeled as DC motors with permanent magnet by:

$$\begin{aligned} u_k &= R_r i_{rk} + C_e \dot{q}_k + L_r \dot{i}_{rk} \\ \tau_k &= C_m i_{rk} + B\dot{q}_k + J_r \ddot{q}_k \end{aligned} . \tag{13}$$

To simplify the model, but with no loss in generality, the same motor model was used for all the joints.

Appendix B
Non-linear Control Algorithm

From the equation of robot dynamics (12) and mathematical model of actuators (13), the system model can be written in the state space. In order to present the model in a controllability canonical form the terminal inductance is neglected ($L_r = 0$). The model in the state space is given in the following form:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \left[\underbrace{\mathbf{D}\left(-\mathbf{h}_0 + \mathbf{J}^T\mathbf{F} - \frac{C_m C_e}{R_r}\dot{\mathbf{q}} \right)}_{\mathbf{f}(\mathbf{x})} \begin{matrix} \dot{\mathbf{q}} \\ \\ \end{matrix} + \underbrace{\frac{C_m}{R_r}\mathbf{D}}_{\mathbf{g}(\mathbf{x})}\mathbf{u} \right], \tag{14}$$

where $\mathbf{D} = (J_r\mathbf{I} + \mathbf{H})^{-1}$ and $\mathbf{I}$ is the unit matrix. As has been already mentioned, by an appropriate choice of the control $\mathbf{u}$:

$$\mathbf{u} = \mathbf{g}(\mathbf{x})^{-1}(\mathbf{v} - \mathbf{f}(\mathbf{x})),$$

the nonlinearities of the system are cancelled, so that the non-linear system reduces to a linear system of double integrators $\ddot{\mathbf{q}} = \mathbf{v}$. If the model were fully accurate, by using the linear regulator:

$$\mathbf{v} = \ddot{\mathbf{q}}^{ref} - \mathbf{K}_D\Delta\dot{\mathbf{q}} - \mathbf{K}_P\Delta\mathbf{q},$$

the poles of the system could be adjusted to the desired values and a desired system dynamics would be achieved. In order to ensure stability of the system, the matrices $\mathbf{K}_P$ and $\mathbf{K}_D$ must be of Hurwitz type. Because of the presence of inaccuracies of the model parameters and external disturbances, the control law should be made more robust. The time-dependent surface $\mathbf{S}(t, \mathbf{x}) = 0$ is defined in the state space:

$$\mathbf{s} = \Delta\dot{\mathbf{q}} + \boldsymbol{\lambda}\Delta\mathbf{q} = 0,$$

where $\boldsymbol{\lambda}$ is the diagonal Hurwitz matrix. When the trajectory of the system in the state space glides over the surface $\mathbf{S}$, the errors of the positions and velocities converge exponentially to zero. In order to ensure a convergent tracking of the trajectory, the control has to satisfy the sliding condition:

$$\frac{1}{2}\frac{d}{dt}\mathbf{s}^T\mathbf{s} = \mathbf{s}^T\dot{\mathbf{s}} \le -\boldsymbol{\eta}\,|\mathbf{s}|, \tag{15}$$

where $\boldsymbol{\eta}$ is a strictly positive constant. The equivalent continuous control which would ensure exponentially convergent tracking of the trajectory if the model were ideally accurate is given by the expression:

$$\mathbf{u}_{eq} = \mathbf{v} - \mathbf{f}(\mathbf{x}) = \ddot{\mathbf{q}}^{ref} - \mathbf{K}_D\Delta\dot{\mathbf{q}} - \mathbf{K}_P\Delta\mathbf{q} - \mathbf{f}(\mathbf{x}).$$

The nonlinear state functions $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are not known but can be estimated by $\hat{\mathbf{f}}(\mathbf{x})$ and $\hat{\mathbf{g}}(\mathbf{x})$ on the basis of a known model. It is assumed that the estimation is bounded by the known functions $\mathbf{F}$ and $\boldsymbol{\beta}$:

$$\left|\mathbf{f} - \hat{\mathbf{f}}\right| \le \mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}), \quad \left|\hat{\mathbf{g}}\mathbf{g}^{-1}\right| \le \boldsymbol{\beta}.$$

In [25], it was shown that the control:

$$\mathbf{u} = \hat{\mathbf{g}}^{-1}\left(\mathbf{u}_{eq} - \mathbf{K}_S sgn(\mathbf{s})\right),$$

with the gain $\mathbf{K}_S$, satisfying:

$$\mathbf{K}_S \ge \boldsymbol{\beta}(\mathbf{F} + \boldsymbol{\eta}) + (\boldsymbol{\beta} - \mathbf{1})\left|\mathbf{u}_{eq}\right|,$$

satisfies the sliding condition (15). In order to eliminate the chattering effect, the nonsmooth function $sgn(\mathbf{s})$ in a thin boundary layer around the surface $\mathbf{S}$ is approximated by the continuous proportional-integral function $\boldsymbol{\rho}(\boldsymbol{\gamma})$:

$$\gamma_i = \frac{s_i}{\Phi_i}, \; i = 1, ..., N;$$

$$\rho(\gamma_i) = \begin{cases} 1 & \text{if } \gamma_i \ge 1 \\ \gamma_i + K_{Ii}\int_{t_{i0}}^{t_i} \gamma_i dt & \text{if } -1 < \gamma_i < 1 \\ -1 & \text{if } \gamma_i \le -1 \end{cases},$$

where Φ_i is the thickness of the boundary layer; $K_{Ii} > 0$ integral gain, and t_{i0} the initial moment when the system state enters the boundary layer [27]. The feedback linearized system in the presence of external disturbances is given by the following vector equation:

$$\ddot{\mathbf{q}} = \mathbf{v} + \mathbf{d},$$

in which $\mathbf{d}$ is a constant disturbance vector. The expanded state vector is defined as:

$$\mathbf{x}_1 = \mathbf{q}, \; \mathbf{x}_2 = \dot{\mathbf{q}}, \; \mathbf{x}_3 = \mathbf{d}.$$

A state estimator of the Luenberger type was implemented. The model of the state and disturbance estimator is given by the following expression:

$$\frac{d}{dt}\hat{\mathbf{x}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}),$$

$$\frac{d}{dt}\begin{bmatrix}\hat{\mathbf{q}} \\ \hat{\dot{\mathbf{q}}} \\ \hat{\mathbf{d}}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{q}} \\ \hat{\dot{\mathbf{q}}} \\ \hat{\mathbf{d}}\end{bmatrix} + \begin{bmatrix}\mathbf{0} \\ \mathbf{I} \\ \mathbf{0}\end{bmatrix}\mathbf{u} + \begin{bmatrix}l_1\mathbf{I} \\ l_2\mathbf{I} \\ l_3\mathbf{I}\end{bmatrix}(\mathbf{q} - \hat{\mathbf{q}}),$$

where $\hat{\mathbf{q}}$ is the vector of the estimated joint angular positions, $\hat{\dot{\mathbf{q}}}$ is the vector of the estimated joint angular velocities, $\hat{\mathbf{d}}$ is the vector of the estimated joint disturbances, and $\mathbf{L}(l_1, l_2, l_3)$ are the corrective factor gains of the estimator. The disturbance estimator behaves as an integral action in the control law, and it contributes to the elimination of the steady-state error. In order to prevent accumulation of the integral action an anti-windup scheme was implemented. The control law used in the simulations is given by the following expression:

$$\mathbf{u} = \hat{\mathbf{g}}(\mathbf{x})^{-1}\left(\ddot{\mathbf{q}}^{ref} - \mathbf{K}_D\Delta\dot{\mathbf{q}} - \mathbf{K}_P\Delta\mathbf{q} - \mathbf{K}_S\boldsymbol{\rho}(\boldsymbol{\gamma}) - \hat{\mathbf{d}} - \hat{\mathbf{f}}(\mathbf{x})\right).$$

References

[1] Y. Breniere and B. Bril, "Development of postural control of gravity forces in children during the first 5 years of walking," *Experimental Brain Research*, vol. 121, no. 3, pp. 255–262, 1998.

[2] T. Keenan and S. Evans, *An introduction to child development*. SAGE Publications Limited, 2009.

[3] K. E. Adolph, B. Vereijken, and P. E. Shrout, "What changes in infant walking and why," *Child development*, vol. 74, no. 2, pp. 475–497, 2003.

[4] G. Cheron, E. Bouillot, B. Dan, A. Bengoetxea, J.-P. Draye, and F. Lacquaniti, "Development of a kinematic coordination pattern in toddler locomotion: planar covariation," *Experimental brain research*, vol. 137, no. 3-4, pp. 455–466, 2001.

[5] V. M. Juricic, D., "Mathematical modeling of a bipedal walking system," in *In Proceedings ASME Winter Annual Meeting*, 1972, pp. 26–30.

[6] M. Vukobratovic, "How to control artificial anthropomorphic systems," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. SMC-3, no. 5, pp. 497–507, 1973.

[7] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa, "Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3989–3994.

[8] N. Perrin, O. Stasse, F. Lamiraux, Y. J. Kim, and D. Manocha, "Real-time footstep planning for humanoid robots among 3d obstacles using a hybrid bounding box," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 977–982.

[9] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2001, pp. 239–246.

[10] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, "A realtime pattern generator for biped walking," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 31–37.

[11] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. 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*, vol. 2. IEEE, 2003, pp. 1620–1626.

[12] K. Hauser, T. Bretl, K. Harada, and J.-C. Latombe, "Using motion primitives in probabilistic sample-based planning for humanoid robots," in *Algorithmic Foundation of Robotics VII*. Springer, 2008, pp. 507–522.

[13] L. Zhang, S. Bi, and D. Liu, "Dynamic leg motion generation of humanoid robot based on human motion capture," in *Intelligent Robotics and Applications*. Springer, 2008, pp. 83–92.

[14] S. Schaal, "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in *Adaptive Motion of Animals and Machines*. Springer, 2006, pp. 261–280.

[15] F. Stulp, E. Theodorou, and S. Schaal, "Reinforcement learning with sequences of motion primitives for robust manipulation," *Robotics, IEEE Transactions on*, vol. 28, no. 6, pp. 1360–1370, 2012.

[16] T. Kulvicius, K. Ning, M. Tamosiunaite, and F. Worgötter, "Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting," *Robotics, IEEE Transactions on*, vol. 28, no. 1, pp. 145–157, 2012.

[17] S. Degallier, L. Righetti, S. Gay, A. Ijspeert, "On the simulation of discontinuous functions," *Autonomous Robots*, vol. 31, no. 2-3, pp. 155–181, 2011.

[18] S. Degallier, A. Ijspeert, "Modeling discrete and rhythmic movements through motor primitives: a review," *Biological cybernetics*, vol. 103, no. 4, pp. 319–338, 2010.

[19] F.L. Moro, N.G. Tsagarakis, D.G. Caldwell, "On the kinematic Motion Primitives (kMPs) - theory and application," *Frontiers in Neurorobotics*, vol. 6, no. 10, 2012.

[20] F.L. Moro, N.G. Tsagarakis, D.G. Caldwell, "A Human-like Walking for the Compliant Humanoid COMAN based on CoM Trajectory Reconstruction from Kinematic Motion Primitives," *IEEE-RAS International Conference on Humanoid Robots*, pp. 364–370, Bled, Slovenia , 2011.

[21] B. Borovac, M. Raković, and M. Nikolić, "Online generation of biped robot motion in an unstructured environment," in *Intelligent Systems: Models and Applications*, ser. Topics in Intelligent Engineering and Informatics, E. Pap, Ed. Springer Berlin Heidelberg, 2013, vol. 3, pp. 93–114.

[22] M. Vukobratović, B. Borovac, M. Raković, V. Potkonjak, and M. Milinović, "On some aspects of humanoid robots gait synthesis and control at small disturbances," *International Journal of Humanoid Robotics*, vol. 5, no. 01, pp. 119–156, 2008.
[23] B. Borovac, M. Nikolić, and M. Raković, "How to compensate for the disturbances that jeopardize dynamic balance of a humanoid robot?" *International Journal of Humanoid Robotics*, vol. 08, no. 03, pp. 533–578, 2011.
[24] M. Vukobratović, H. Herr, B. Borovac, M. Raković, M. Popovic, A. Hoffmann, M. Jovanović, and V. Potkonjak "Biological principles of control selection for a humanoid robot's dynamic balance preservation." *International Journal of Humanoid Robotics*, vol. 05, no. 04, pp. 639–678, 2008.
[25] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice hall New Jersey, 1991, vol. 1, no. 1.
[26] A. Radke and Z. Gao, "A survey of state and disturbance observers for practitioners," in *American Control Conference, 2006*, 2006.
[27] T. V. Nguyen, Q. Ha, and H. T. Nguyen, "A chattering-free variable structure controller for tracking of robotic manipulators," in *Proceedings of the 2003 Australasian Conference on Robotics & Automation, Brisbane, Australia (December 1–3, 2003)*, 2003.
[28] V. Potkonjak, M. Vukobratović, K. Babković, and B. Borovac, "General model of dynamics of human and humanoid motion: feasibility, potentials and verification," *International Journal of Humanoid Robotics*, vol. 3, no. 01, pp. 21–47, 2006.
[29] P. Song, P. Kraus, V. Kumar, and P. Dupont, "Analysis of rigid-body dynamic models for simulation of systems with frictional contacts," *Transactions-America Society of Mechanical Engineers Journal of Applied Mechanics*, vol. 68, no. 1, pp. 118–128, 2001.
[30] S. Goyal, E. N. Pinson, and F. W. Sinden, "Simulation of dynamics of interacting rigid bodies including friction i: General problem and contact model," *Engineering with computers*, vol. 10, no. 3, pp. 162–174, 1994.
[31] S. Goyal, E. N. Pinson, and F. W. Sinden, "Simulation of dynamics of interacting rigid bodies including friction ii: Software system design and implementation," *Engineering with computers*, vol. 10, no. 3, pp. 175–195, 1994.
[32] J. Turner, "On the simulation of discontinuous functions," *Journal of applied mechanics*, vol. 68, no. 5, pp. 751–757, 2001.

Mirko Raković was born in 1982 in Sremska Mitrovica, Serbia. He received a Ph.D. degree in Mechatronics and M.Sc. degree in Electrical Engineering and Computer Science from the Faculty of Technical Sciences Novi Sad in 2013 and 2005 respectively. Since then he has been working as a research associate and teaching assistant at the Chair of Mechatronics, Robotics and Automatization of the Faculty of Technical Sciences Novi Sad. His research interests include robotics, bipedal locomotion, robot modelling and control, embedded systems and programming.

Branislav Borovac was born in Leskovac, Serbia, 1951. He received the M.Sc. and Ph.D. degrees in Mechanical Engineering from the University of Novi Sad in 1982 and 1986 respectively. His research interests include biped locomotion, humanoids, cognition, robot modelling and control, industrial robotics, sensors and sensor information integration, force sensors and their use in contact tasks, assembly, mechatronics, product design, and flexible systems.

Milutin Nikolić was born in 1984 in Novi Sad, Serbia. He received an MS degree in Mechatronics at the Faculty of Technical Sciences (FTN), Novi Sad. He is currently a research associate and Ph.D. student at the Chair of Mechatronics, Robotics and Automatization at the FTN. His field of research includes robotics, bipedal gait, mechatronics and computer vision.

Srđan Savić was born in 1987 in Novi Sad, Serbia. He received an MS degree in Mechatronics at the Faculty of Technical Sciences (FTN), Novi Sad. He is currently a research assistant and Ph.D. student at the Chair of Mechatronics, Robotics and Automatization at the FTN. His field of research includes robotics, non-linear control, mechatronics and fractional calculus.