Control Algorithm for Stable Walking of Biped Robots

Konstantin Kondak, Günter Hommel
Technische Universität Berlin, Institut für Technische Informatik und Mikroelektronik, Germany

ABSTRACT

This paper deals with the stable walking of biped robots. The presented control algorithm enables a biped to perform stable walking without using any precomputed trajectories. The algorithm merges gait trajectory generation and control, and can be used for global control, for local control along an existing trajectory as well as for online computation of gait trajectories for stable walking. The inputs for the algorithm are a few parameters such as walking speed and step size. The performance of the algorithm is demonstrated by simulation.

1 INTRODUCTION

The challenge in movement control of biped robots is to ensure overall stability or balance of the system during task execution. Consideration of the overall stability of a biped, with mass properties equivalent to those of a human, is necessary even for slow walking.

In most of the existing elaborated biped systems, e.g. [1, 2], the walking control is performed on two levels: computation of the gait trajectories and local control along them. The overall stability or balancing should be considered on both levels.

On the first level, the synergy method [3] and methods based on optimization [4] are often used.

In case of synergy methods, the trajectories for most of the joints are generated e.g. from recorded human movement, and the trajectories for the few remaining joints are computed in relation to overall stability of the robot. Usually, the movements of the trunk in frontal and sagittal planes are considered unknown. Computation of the unknown trajectories requires the solution of a boundary value problem, and can only be performed using iterative numerical methods.

The methods based on non-linear optimization compute the movements of the robot by minimizing a cost function. This cost function specifies the movement properties which are central to the given task, e.g. minimal time and/or minimal energy consumption. The dynamical equations and stability conditions are incorporated in the optimization problem as equality and inequality constraints. Even the solution for a simple model can be found only numerically and requires high computational effort.

The controllers on the second level generate precomputed trajectories in the actuators and balance the biped in the small region of the nominal precomputed movement.

In this paper, a control algorithm which merges these two levels is presented. This algorithm does not require any numerical iterative methods and has minimal computational effort.

The presented control algorithm was developed for steering an exoskeleton by the force imposed by the human in it. Smooth force steering requires a reaction time of approx. $1\,ms$, which makes the use of even fast iterative numerical computations of stable gait problematic. The number of possible human movements in the exoskeleton is vast and an offline precomputation for even a single robot is problematic. Furthermore, by switching from one precomputed trajectory to another, the movement can become unstable because the distance between two trajectories could exceed the operation region of the local controller.

2 SHORT DESCRIPTION OF THE CONTROL ALGORITHM

The biped robot is modeled as a chain of seven rigid bodies: both feet, lower legs, upper legs and trunk, as shown in Fig. 1.

The dynamical equations were derived using Kane's formalism [5], and have the following form:

$$\mathbf{M}(\mathbf{q})\,\dot{\omega} = \mathbf{f}(\mathbf{q},\omega) + \mathbf{T} \tag{1}$$

where $\mathbf{q} = (q_1,\ldots,q_6)^T$ is the vector of generalized coordinates, which are angles in ankles, knees, hip joints, and $\omega = (\omega_1,\ldots,\omega_6)^T$ is the vector of corresponding generalized velocities. The matrix function $\mathbf{M}(\mathbf{q})$ takes into account the mass distribution, and the vector function $\mathbf{f}(\mathbf{q},\omega)$ describes the influence of both inertial forces and gravity. The elements of the vector $\mathbf{T}$ are generalized forces applied to the system. For the model considered, these are the torques in the joints. The dot denotes the time derivative in a Newtonian reference frame.

The kinematic equations for the model are obviously:

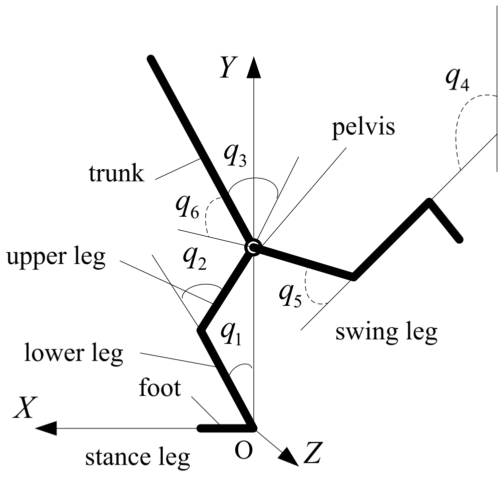

Figure 1: The model of the biped robot.

$$\dot{\mathbf{q}} = \omega \tag{2}$$

The scheme of the control algorithm is shown in Fig. 2.

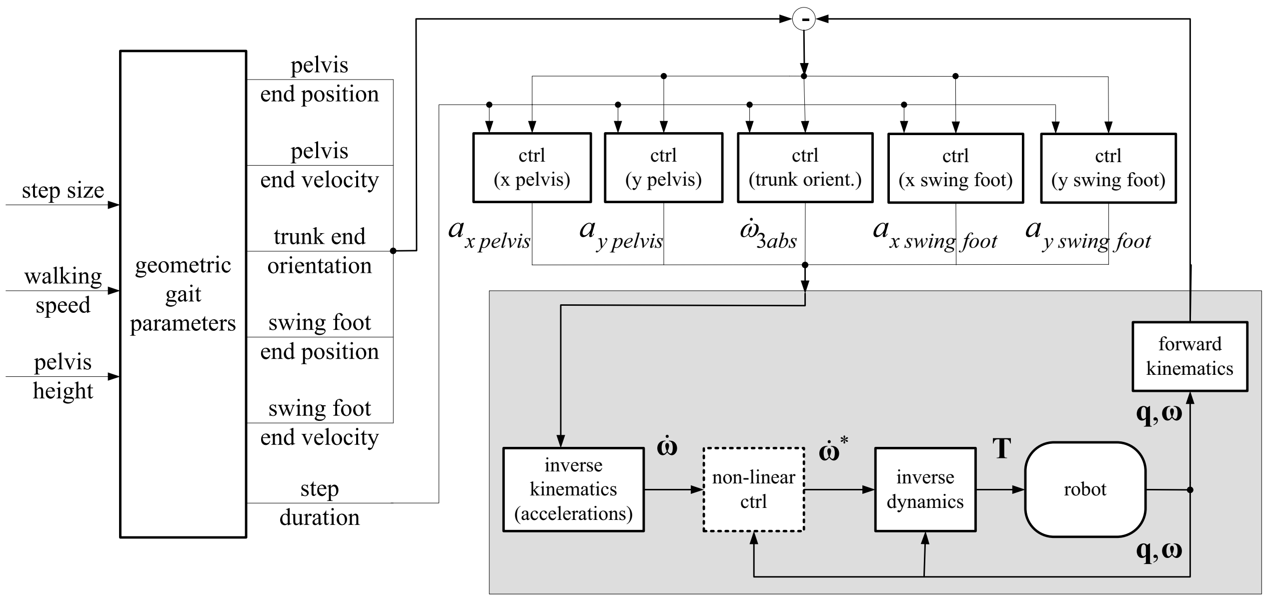

Figure 2: Scheme of the algorithm.

Walking is specified by a few task parameters which are the inputs of the control algorithm (left in Fig. 2): step size, walking speed and desired height of the pelvis. From these task parameters, the geometric gait parameters are calculated in block *geometric parameters*. For the model in Fig. 1

the following geometric gait parameters can be used: end position and velocity of the pelvis, trunk orientation, end position and velocity of the swing foot and step duration. End means the end time of each step – time point where the stance leg becomes the swing leg and vice versa. This block serves to facilitate the usage of the algorithm and is described in sec. 5.
The block *inverse kinematics (accelerations)* computes the acceleration vector $\dot{\omega}$ from the accelerations of the geometric gait parameters as shown in Fig. 2. For that a system of linear equations should be solved, which is obtained by double differentiation of the forward kinematics equations. The solution can be performed symbolically therefore the formulas for calculation of the elements of $\dot{\omega}$ are available.
For a moment we assume that the dotted block *non-linear ctrl* has a transfer function equal to 1 and $\dot{\omega} = \dot{\omega}^*$.
The block *robot* represents the real biped or its non-linear model.
The block *inverse dynamics* computes the joint torques $\mathbf{T}$ from the given acceleration vector $\dot{\omega}^*$ and the actual system state $(\mathbf{q}, \omega)$ using dynamical eq. (1). This block linearizes and decouples the non-linear system which describes the biped, so that the blocks *robot* and *inverse dynamics* are equivalent to the six independent double integrators, one for each joint.
The block *forward kinematics* computes the actual values for geometric gait parameters from the actual system state $(\mathbf{q}, \omega)$.
Due to the assumption that $\dot{\omega} = \dot{\omega}^*$, the shaded area in Fig. 2 can be considered as five independent double integrators, which describe the behavior of each of the geometrical gait parameters. These parameters are independently controlled by the blocks *ctrl (...)*.
Until now, the control scheme looks similar to the widely used approach for decoupling, linearization and control for non-linear mechanical systems, especially arm manipulators (see e.g. [6]). The application of the described scheme for global control of stable walking of a biped involves the following two issues:

- to produce a periodical gait the blocks *ctrl (...)* should force the geometrical parameters to reach its end values *at the same time*
- the overall stability of the robot should be ensured during walking

The main contribution of this study is the elaboration of these two problems.
As shown in Fig. 2, the first problem is solved by means of a controller, which is designed using the well known results for state reachebality in linear systems in finite time. The control law for blocks *ctrl(...)* is given in sec. 3.
For achieving overall stability, the concept of Zero Moment Point (ZMP) [7] is used. The overall stability is achieved by modifying the acceleration vector $\dot{\omega}$ in the block *non-linear ctrl*. This modification will be performed by projecting the vector $\dot{\omega}$ onto the plain, which is denoted by the authors as *ZMP-plane*. The projection operator requires the solution of a system of linear equations with the dimension one less than the dimension of the vector ω. This means that computations in the block *non-linear ctrl* are not expensive and the method can be easily applied to models with more dimensions. This block is described in sec. 4.
After setting the described task parameters the algorithm produces a stable, periodical walking pattern corresponding to these parameters, which could be modified during the robot movement.

3 THE TIME DEPENDENT CONTROLLER FOR GEOMETRICAL PARAMETERS

To produce a periodical gait, all of the blocks *ctrl(...)* in Fig. 2 should force the geometrical parameters to reach its end values at the same time. Due to linearization and decoupling of the system, each

of the geometrical parameters has the behavior of a double integrator.
The computation of the input $u(t)$ which quides the linear time invariant system from the actual state $\mathbf{x}_0$ into some state $\mathbf{x}_1$ at the given time t_1 can be performed as follows:

$$u(t) = \mathbf{B}^T \Phi^T(t) \left(\mathbf{W}^T(t_1)\mathbf{W}(t_1)\right)^{-1} \mathbf{W}^T(t_1)\left(\mathbf{x}_1 - \Phi(t_1)\mathbf{x}_0\right) \tag{3}$$

where $\mathbf{W}(t_1)$ is the controllability grammian:

$$\mathbf{W}(t_1) = \int_{t_0}^{t_1} \Phi^{-1}(t)\mathbf{B}(t)\mathbf{B}^T(t)\left(\Phi^{-1}\right)^T(t)dt$$

$\Phi(t) = \mathcal{L}^{-1}\left\{(s\mathbf{I} - \mathbf{A})^{-1}\right\}$ is the transition matrix, **A** the system matrix and **B** the input matrix of the linear system. After the substitution the integration variable t with $\tau = t - t_1$ and using the property of the transition matrix $\Phi^{-1}(t) = \Phi(-t)$, the controllability grammian $\mathbf{W}(t_1)$ can be rewritten as:

$$\mathbf{W}(\Delta t) = \int_{-\Delta t}^{0} \Phi(-\tau)\mathbf{B}(\tau)\mathbf{B}^T(\tau)(\Phi)^T(-\tau)d\tau \tag{4}$$

where $\Delta t = t_1 - t_0$ is the remaining time to reach $\mathbf{x}_1$ from the actual state $\mathbf{x}_0$.
The derivation of the control law (3) can be found in each advanced control theory book, e.g. [8]. After inserting in (3) and (4) the system matrices of double integrator $\mathbf{A} = (0,1;0,0)$ and $\mathbf{B} = (0,1)^T$with the state vector $\mathbf{x} = (x, \upsilon)^T$composed of position x and velocity υ one obtains the time dependent control law for each of the blocks *ctrl(...)* in Fig. 2:

$$u(\Delta t) = a_i(\Delta t) = \frac{6(x_1 - x_0) - 2\Delta t(2\upsilon_0 + \upsilon_1)}{\Delta t^2} \tag{5}$$

In the vicinity of the end state $\mathbf{x}_1 = (x_1, \upsilon_1)^T$ Δt goes to 0 and eq. (5) can not be used. In this case, a small positive double value Δt_{min} can be used instead of Δt or the control law (5) can be replaced, e.g. by a PID-controller. In the simulations presented in sec. 6 the first approach with $\Delta t_{min} = 0.01$ has been used.

4 MODIFICATION OF THE ACCELERATION VECTOR $\dot{\omega}$

The stability condition for the movement in the sagittal plane will be defined in this work as:

$$x_{min} \leq x_{zmp} \leq x_{max} \tag{6}$$

where x_{min} and x_{max} are two margins of the foot print and x_{zmp} is the position of ZMP. This means that during the movement the ZMP remains within the foot print.
It could be shown (see e.g. [7, 9]) that the expression for x_{zmp} has the form:

$$x_{zmp} + \alpha_0 = \sum_{i=1}^{6} \alpha_i \dot{\omega}_i \tag{7}$$

where $\alpha_0, \ldots, \alpha_6$ are non-linear functions depending on generalized coordinates **q** and velocities ω. At each moment, the acceleration of the system $\dot{\omega}$ can be changed arbitrarily by application of corresponding torques **T** in the joints (according to eq. (1)). By contrast, the changes in velocities ω correspond to actual accelerations, and the changes in coordinates **q** correspond to actual velocities (double integrator behavior). This allows the coefficients $\alpha_0, \ldots, \alpha_6$ in eq. (7) at each moment to be considered constants and the accelerations $\dot{\omega}$ variables with arbitrary values. Therefore, for each value of x_{zmp} eq. (7) describes a plane in the acceleration space at each moment. This plane is called

the *ZMP-plane*. Of course, the position and orientation of this plane are changing in time. This ZMP-plane has a clear physical meaning: choosing the actual acceleration $\dot{\omega}$ belonging to it, we guarantee that the system moves at this moment of time in such a way that the x-coordinate of ZMP will be equal to x_{zmp}.
The block *non-linear ctrl* ensures the satisfaction of the stability condition (6) and is implemented using the following algorithm:

```
1. calculate $x^*_{zmp}$, corresponding to $\dot{\omega}$
2. if($x^*_{zmp} < x_{min}$ or $x^*_{zmp} > x_{max}$)
   then
      $x^*_{zmp} = x_{min}$ or $x^*_{zmp} = x_{max}$
      $\dot{\omega}^* = projection(\dot{\omega})$
   else
      $\dot{\omega}^* = \dot{\omega}$
```

The operator $projection(*)$ in step 2 can be implemented in different ways. It is important that the resulting acceleration vector $\dot{\omega}^*$ lies in the ZMP-plane which corresponds to x_{zmp} satisfying condition (6). Three different definitions of this operator (see Fig. 3) were investigated.
For the sake of clarity in Fig. 3, only two of six dimensions of the acceleration space are shown. In the first definition the length of the origin vector $\dot{\omega}$ is changed in such a way that its tip lies in the ZMP-plane. The resulting vector is denoted as $\dot{\omega}^*_1$. In the second definition, the end of the resulting vector, denoted as $\dot{\omega}^*_2$, is given by point B, which is the crossing point of the perpendicular from the end of the origin vector $\dot{\omega}$ (point A) to the ZMP-plane. In the third definition, five coordinates of the origin vector $\dot{\omega}$ remain unchanged and only one is adjusted. The resulting vector is denoted as $\dot{\omega}^*_3$. The simulation of different movements has shown that the second definition of the operator $projection(*)$ leads to the best movement performance.

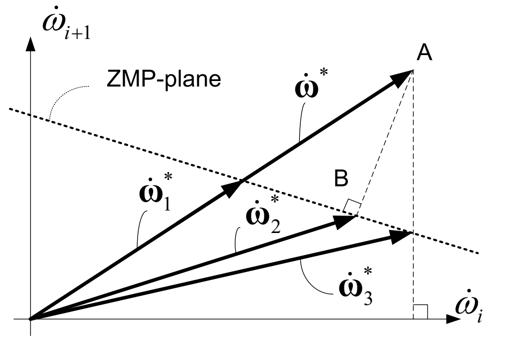

Figure 3: Three definitions for operator $projection(*)$.

The implementation of each of these three operators requires the solution of a corresponding system of linear equations with dimension less than the dimension of the vector ω.
By active operator $projection(*)$ the geometrical parameters do not behave like double integrators and the periodical movement generated by blocks *ctrl(...)* is disturbed. The operator $projection(*)$ should try to redistribute this disturbance over all dimensions of the acceleration space, so that each particular coordinate has only a small disturbance which can be corrected when the operator $projection(*)$ is not active.

5 COMPUTATION OF GEOMETRICAL GAIT PARAMETERS

The rules for computation of geometrical gait parameters (see Fig. 2) have an impact on the period of time in which the operator $projection(*)$ is active. Unfortunately, there is no formal way to find camputation rules that minimize the active time of $projection(*)$. To find appropriate rules the authors used the following approach: The transfer function of the block *non-linear ctrl* was set to 1 and the position of the ZMP was observed during the simulation of walking with different rules. In this simulation it was assumed that the robot does not rotate about the foot edge, so that the ZMP could leave the foot print without the robot falling down. The rules have to be varied in order to force the ZMP to remain within the foot print as long as possible.

Good results were achieved with the following rules: $pelvis\,end\,position_x = step\,size/2$, $pelvis\,end\,position_y = pelvis\,height$, $pelvis\,end\,velocity_x = 1.5 * walking\,speed$, $trunk\,end\,orientation = 0.23\,rad$ (statical equilibrium for $pelvis_x = 0$, $pelvis_y = pelvis\,height$ and $x_{zmp} = 0$, both feet are on the floor side by side), $swing\,foot\,end\,position_x = step\,size$, $swing\,foot\,end\,position_y = 0$, $swing\,foot\,end\,velocity_x = 0$ and $swing\,foot\,end\,velocity_y = 0$. All specifications refer to the reference frame shown in Fig. 1, *end* means the end of the step.

6 SIMULATION RESULTS

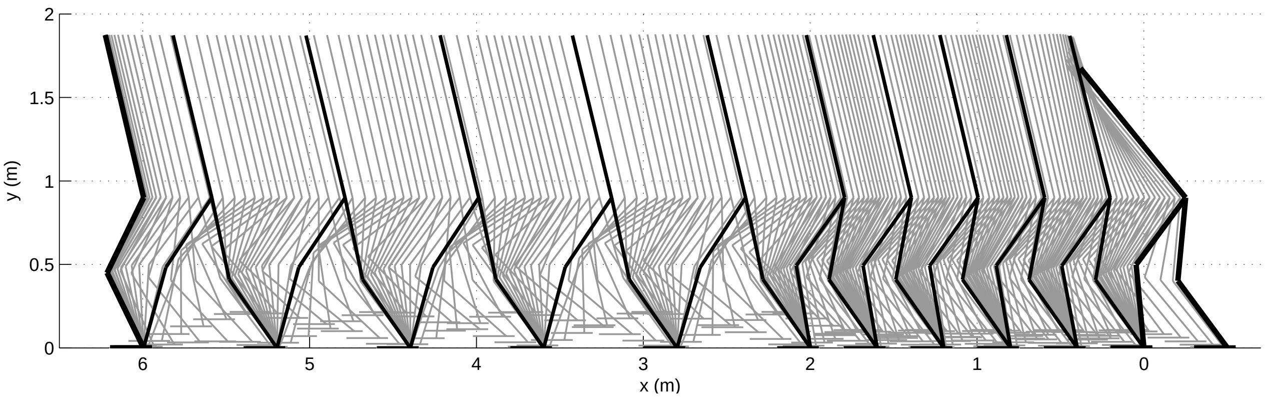

Figure 4: Resulting movement for walking with different parameters

To demonstrate the performance of the presented control algorithm, the following computer simulation experiment was performed: The biped was standing in some arbitrarily chosen initial state (the center of mass (CM) was within a footprint, all generalized velocities were equal to 0). The biped had to start walking with the following task parameters: step size - $0.4\,m$, walking speed - $0.4\,m/s$ and pelvis height - $0.9\,m$. After walking forabout $5\,s$ the biped had to double the step size and the walking speed. After walking with these new parameters for further $5\,s$ the step size was set to 0 and the biped had to stop. The mass of upper and lower legs was set to $5\,kg$ each; the mass of the trunk to $50\,kg$ (total mass $70\,kg$); the length of the upper and lower legs was set to $0.5\,m$ and the length of the trunk junk to $1\,m$. All body parts were modeled as solid cylinders with uniform mass distribution.
The diagram in Fig. 4 illustrates the resulting walking movement. The configurations where the stance foot is changed as well as the start (right) and the end (left) configurations are marked with bold. Fig. 5-7 reveal the details of this movement. In each of these Fig., two intervals corresponding to walking with different task parameters can be seen. Each interval starts with a settling process, which lasts one step interval, after that the periodic change of the values can be seen in each Fig.. The coordinates in all Fig. are given relative to the ankle of the stance foot (point O in Fig. 1). At the end of each step (in this example the step duration is equal to $1\,s$) the swing foot becomes the stance foot and vice versa without double support phase and without ballistic flying. At the time the swing foot touches the floor, its absolute velocity is equal to 0.
The trajectories for the x-coordinates of ZMP and Center of Mass (CM) are shown in Fig. 5. The x-coordinate of the ZMP remains within the chosen range (within the foot print) $[-0.05\,m; 0.2\,m]$ during the whole movement ($x = 0$ is in point O, see Fig. 1). After the biped has stopped, both ω and $\dot{\omega}$ are zero, and the x-coordinates of ZMP and CM should coincide. Exactly this can be seen in Fig. 5 at the end of the movement after about $11\,s$.
The horizontal velocities of the pelvis and CM are shown in Fig. 6. After the settling process in each interval, the average values of these velocities are equal and are in accordance with the specified walking speeds.

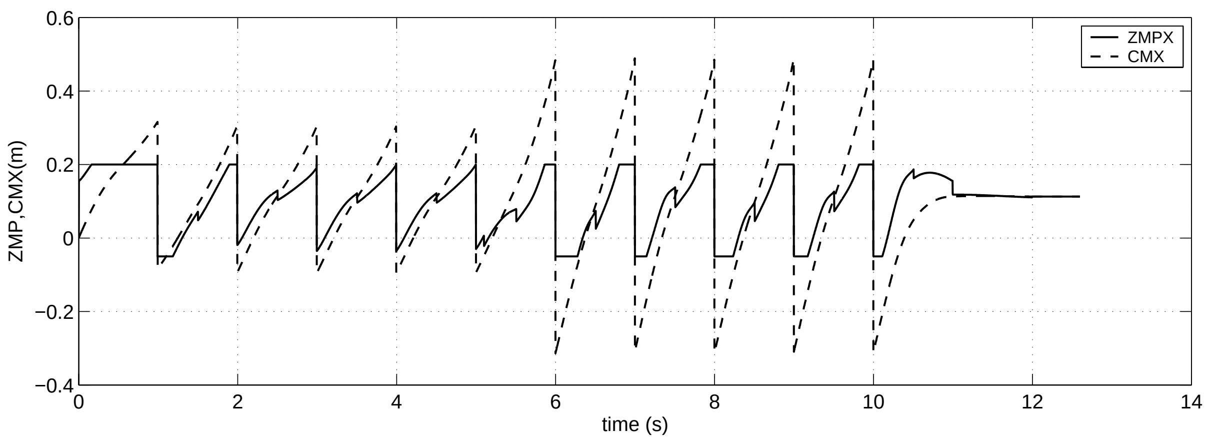

Figure 5: Trajectories for ZMP and CM

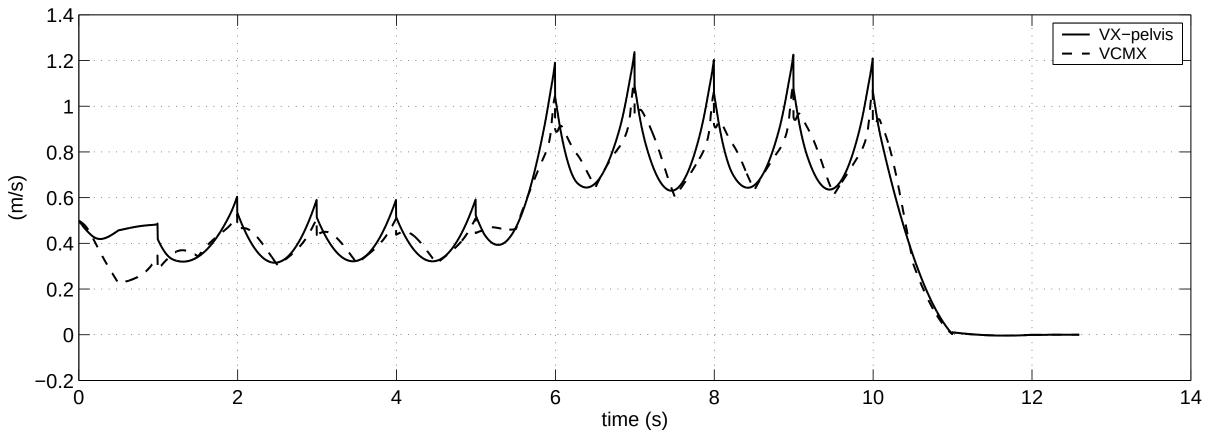

Figure 6: Horizontal velocities of the pelvis and CM

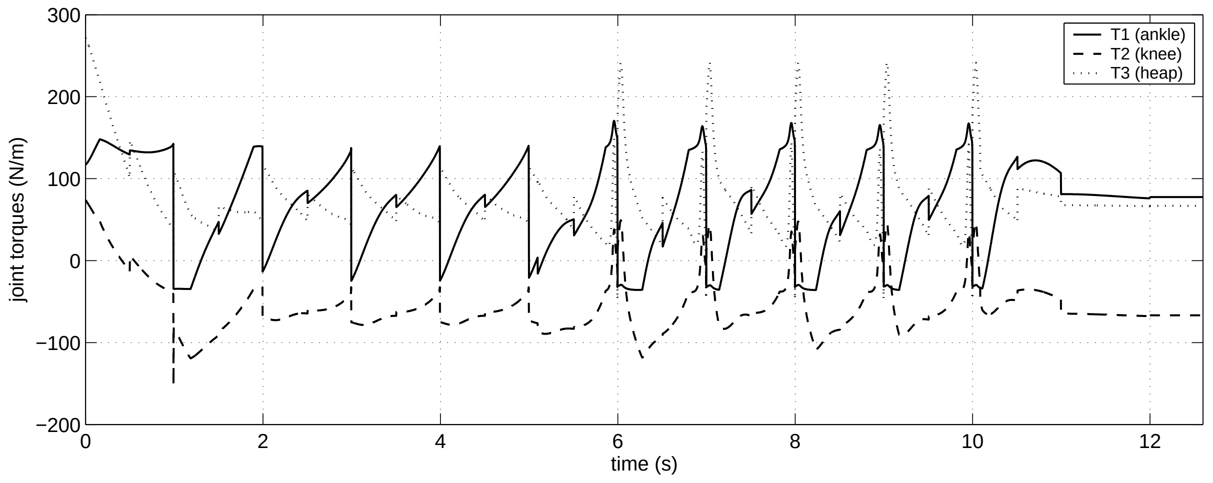

Figure 7: Torques in joints 1,2,3 **(see Fig. 1)**

Fig. 7 shows the applied torques in the joints of the stance leg.
In this study walking without double support phase is considered. After the swing foot reaches the floor it becomes immediately the stance foot and vice versa. This case is more complicated for balancing than the case when a double support phase is present. During the double support phase,

the range allowed for ZMP is bigger (convex hull of two foot prints instead of one) and the biped can endure a larger acceleration to correct its movement.
In several published studies the control of the ZMP position is used to achieve overall stability of a biped. To the authors' opinion, this approach has drawbacks due to the following considerations: Prescribing the trajectory of the ZMP position reduces the number of independent degrees of freedom of the robot. Since the trajectory of the ZMP position does not directly correspond to the tasks of the robot, other task relevant value, e.g. trunk orientation or pelvis velocity, can not be controlled independently. Furthermore, the ZMP position linearly depends on the accelerations of the generalized coordinates, which in turn, depend on the applied torques. This means that the ZMP position can be changed arbitrary and any time by applying appropriate torques.

7 CONCLUSION

It was shown that the trajectory generation for stable walking of a biped and the control along these trajectories could be merged into a single algorithm in form of a feedback controller. The performance of the algorithm was demonstrated by simulation. As mentioned above, the application field for the algorithm ranges from global control to online trajectories computation.
The authors were surprised that stable walking with changeable parameters could be achieved using a feedback controller without any compute-intensive "intelligent" calculations.
The presented version of the algorithm does not consider the limitation of the torques in actuators, which is very important for practical applications. The authors are working on this problem.

References

[1] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "Development of honda humanoid robot," in *Proceedings of the IEEE Int. Conf. on Robotics & Automation*, pp. 1321–1326, 1998.

[2] M. Gienger, K. Löffler, and F. Pfeiffer, "Towards the design of a biped jogging robot," in *Proceedings of the IEEE Int. Conf. on Robotics & Automation*, pp. 4140–4145, 2001.

[3] M. Vukobratović, B. Borovac, D. Šurla, and D. Stokić, *Biped Locomotion. Dynamics, Stability, Control and Application.* Springer, 1990.

[4] F. Bahrami, R. Riener, M. Buss, and G. Schmidt, "Optimal trajectories for paraplegic patients raising from a chair by means of FES: preliminary results," in *Neuroprosthetics, from basic research to clinical application* (A. Pedotti, F. Ferrarin, J. Quintern, and R. Riener, eds.), pp. 285–292, Springer, 1996.

[5] T. Kane and D. Levinson, *Dynamics OnLine: Theory and Implementation with Autolev*. Kane Dynamics, Inc., 2000.

[6] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. John Wiley & Sons, 1989.

[7] M. Vukobratović, B. Borovac, and D. Surdilovic, "Zero-moment point – proper interpretation and new applications," in *Proceedings of the IEEE-RAS Int. Conf. on Humanoid Robots*, pp. 237–244, 2001.

[8] G. Ludyk, *Theoretische Regelungstechnik 1*. Springer, 1995.

[9] K. Kondak and G. Hommel, "Control and online computation of stable movement for biped robots," in *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003, accepted for publication.