Control and Online Computation of Stable Movement for Biped Robots

Konstantin Kondak, Günter Hommel

Technische Universität Berlin,
Institut für Technische Informatik und ME,
Einsteinufer 17, Sekr.: EN10, 10587 Berlin
{kondak,hommel}@cs.tu-berlin.de

***Abstract*— The presented algorithm enables a biped to perform a stable movement to a defined goal without using any precomputed trajectories. The algorithm merges the trajectory generation and the control along it, and can be used for global control, for local control along an existing trajectory, as well as for online computation of trajectories for stable movement. The algorithm is based on a decoupling of the non-linear model and changing the steering torques to account for both the overall stability of the biped and for achieving the goal. The algorithm is applied to the model of a biped moving in sagittal plane. By specifying the goal as an arbitrary pelvis position the biped can perform movements such as sitting down or standing up. The performance of the algorithm is demonstrated in a simulation.**

I. Introduction

Consideration of the overall stability or the balancing of a biped, which has the mass properties equivalent to those of a human, is necessary even for simple and slow movements during the task execution. Otherwise the biped will rotate over the foot edge and collapse even if the trajectories in actuators do not deviate from the nominal ones.

In most of the existing elaborated biped systems, e.g. [1], [2], the movement control is divided in two levels: computation of the trajectories for a given task, 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 movements, 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 a solution of the 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 consumptions. 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 reproduce precomputed trajectories in the actuators and balance the biped in the small region of the nominal precomputed movement.

In this paper, an algorithm which merges these two levels is presented. It can be used in three ways: as a global controller which can stably steer the biped to the arbitrary goal in the reachable work space; as local controller with a large operation range; and as an algorithm for the online computation of trajectories of balanced movement. In this paper, only the movements without changing a stance foot are considered.

The problem, considered in this paper is similar to the problem of postural adjustment of bipeds, s. e.g. [5].

The presented algorithm was developed for steering an exoskeleton by the force imposed by the human in it. In this application the robot must be able to perform movements which are similar to those of the human, which could range widely. The number of possible movements here 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.

II. Problem Formulation

A. Model of the biped Robot

The biped robot is modelled as a chain of four rigid bodies: foot, shank, thigh and trunk, as shown in Fig. 1. The dynamical

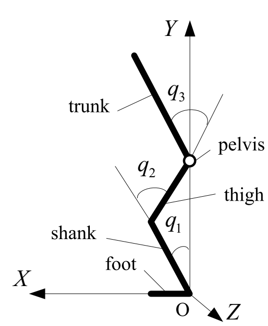

Fig. 1. The model of the biped robot.

equations were derived using Kane's formalism [6], and have the following form:

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

where $\mathbf{q} = (q_1, q_2, q_3)^T$ is the vector of generalized coordinates, which are angles in ankle, knee, hip joints, and $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)^T$ is the vector of the corresponding generalized velocities. The matrix function $\mathbf{M}(\mathbf{q})$ takes into account the mass distribution and the vector function $\mathbf{f}(\mathbf{q}, \boldsymbol{\omega})$ describes the influence of both, the inertial forces and the 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 derivate in a Newtonian reference frame.

The kinematical equations for the model are obviously:

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

The proposed control algorithm will be described using the model shown in Fig. 1. This means that only the movements in the sagittal plane of the robot (XY-plane) will be considered. As will be explained in sec. III, the algorithm can be easily extended for more complicated 3D models.

B. Stability Condition

For the formulation of the stability condition, the concept of the Zero Moment Point (ZMP) [7], [8] is used. The original definition of the ZMP has been slightly modified, so as to incorporate the definition of the imaginary ZMP [7]:

Definition 1: The ZMP is an imaginary point where the resulting ground reaction force *should be applied*, so that the resulting torque imposed on the foot becomes zero.
This definition coincides with the definition of the foot rotation indicator (FRI) introduced in [9]. In the authors' opinion the above-mentioned modification of the ZMP definition does not justify the coining a new term for a concept which has been discussed in relevant literature for over 30 years (for detailed discussion of the concept itself, see [7] and [9]).

The ZMP-concept and overall stability condition are illustrated in Fig. 2. The forces which are exerted on the foot are shown in Fig. 2 (a). It is assumed that the foot does not move relatively to the floor. The influence from the side of the robot is replaced with two forces F_x, F_y and with the torque M_z. From the ground, the foot experiences a pressure distribution p and sheer friction force distribution f_s. In most practical situations, F_x and f_s can be disregarded. The distributed pressure is always applied to the foot sole in one direction (the sole does not stick to the ground) and can be replaced with a single resulting force F_r applying to some point within the foot print, s. Fig.2 (b). The forces F_y and F_r are equal and constitute a force couple with a torque equal to $x_{zmp}F_r$, where x_{zmp} is the distance between the action lines of these forces.

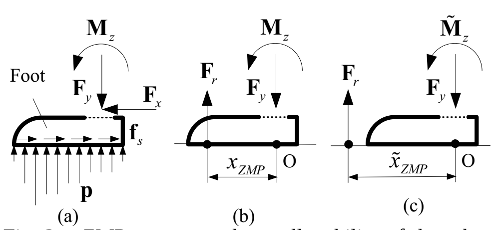

Fig. 2. ZMP concept and overall stability of the robot.

If the force couple formed by F_y and F_r cancels the torque M_z, the foot does not rotate around the edge. In this case, the ZMP lies within the foot print. If the torque generated by force couple F_y and F_r cannot compensate for the torque M_z, the foot begins to rotate around its edge. In this case, according to definition 1, ZMP leaves the region given by the foot print, see Fig. 2 (c).

The rotation around the edge of the foot does not necessarily mean that the biped is going to collapse. Theoretically this rotation could still be controlled by means of inertial forces generated by accelerated movements of the body parts, or stopped by reconfiguration of the kinematical structure of the system, e.g. additional support with the other foot. But the question is whether such a movement has any advantages. Intuitively, it is clear that control of the system during rotation around the edge of the foot requires more effort. It should also be mentioned here that it is not obvious in which manner the human does in fact move. Even in case of running, during the single support phase the ZMP could be in- or outside of the foot print. To the authors' knowledge this question remains still open.

In this work the stable movement will be defined as follows:

Definition 2: The movement of the biped will be called stable if the ZMP lies within the foot print.
The stability condition for the movement in the sagittal plane can be written therefore as:

$$x_{min} \leq x_{zmp} \leq x_{max} \tag{3}$$

where x_{min} and x_{max} are two margins of the foot print and x_{zmp} is given by:

$$x_{zmp} = -\frac{M_z}{F_y} \tag{4}$$

It is worth mentioning that the violation of the movement stability condition (3) could be caused by both a large torque M_z (e.g. the center of mass is moving too fast or is displaced too far forwards/backwards), and a small force F_y (e.g. the center of mass is moving with too high acceleration downward).

III. Control Algorithm for Stable Movement

A. The Idea

The scheme of the control algorithm is shown in Fig. 3. The movement goal is specified as x,y coordinates of a reference point, which can be chosen arbitrarily. In this study, the pelvis was taken as a reference point (see Fig. 1). The block *kinematical transformation* computes the joint coordinates $\mathbf{q}$ which correspond to the specified position of the reference point. For the model in Fig. 1, this position is uniquely defined by the angles q_1 and q_2. To define the angle q_3 – trunk orientation – the equation for the position of the center of mass (CM) is used:

$$x_{cm} = g(\mathbf{q}) \tag{5}$$

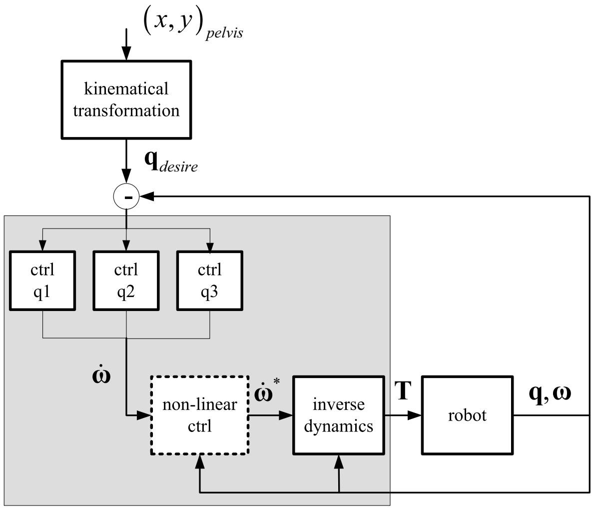

Fig. 3. Scheme of the algorithm.

where x_{cm} is a x-coordinate of the CM and $g(*)$ is a non-linear function in $\mathbf{q}$. If x_{cm} is specified, the trunk orientation q_3 can be easily computed from eq. (5). It is reasonable to set x_{cm} in the middle of the range, specified for x_{zmp}:

$$x_{cm} = (x_{min} + x_{max})/2 \tag{6}$$

If the dotted block *non-linear ctrl* has a transfer function equal to 1 which means that $\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}^*$, the control scheme becomes similar to the widely used approach for decoupling, linearization and control for non-linear mechanical systems, especially arm manipulators (see e.g. [10]).

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 $\dot{\boldsymbol{\omega}}^*$ and the actual system state $(\mathbf{q}, \boldsymbol{\omega})$ using dynamical eq. (1). This block linearizes and decouples the non-linear system which describes the biped, so that blocks *robot* and *inverse dynamics* are equivalent to the three independent double integrators. The movement of each of these double integrators could be independently controlled in different ways, e.g. with a PD or PID controller. This three controllers are denoted in Fig. 3 as $ctrl\, q_1$, $ctrl\, q_2$, $ctrl\, q_3$.

The application of the described scheme for global control of a biped requires accounting for stability condition (3). The incorporation of this stability condition into the control algorithm is the main contribution of this study. Accounting for the stability condition (3) will be achieved by modifying the accelaration vector $\dot{\boldsymbol{\omega}}$ in the block *non-linear ctrl*. As shown in sec. III-B, this modification requires the solution of a system of linear equations with the dimension one less than the dimension of the vector $\boldsymbol{\omega}$. 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.

B. The Control Algorithm

As mentioned in the previous section, the novelty of the proposed algorithm is the modification of the acceleration vector $\dot{\boldsymbol{\omega}}$ obtained initially as an input signal for three double integrators. This modification is performed by projecting the vector $\dot{\boldsymbol{\omega}}$ onto two planes, which are denoted by the authors as the *ZMP-* and *CM-plane*. The notions of these two planes are presented below.

After inserting the explicit formulas for F_y and M_z, eq. (4) can be rewritten as follows:

$$x_{zmp} + \alpha_0 = \alpha_1 \dot{\omega}_1 + \alpha_2 \dot{\omega}_2 + \alpha_3 \dot{\omega}_3 \tag{7}$$

where $\alpha_0, \ldots, \alpha_3$ are non-linear functions depending on generalized coordinates $\mathbf{q}$ and velocities $\boldsymbol{\omega}$. At each moment, the acceleration of the system $\dot{\boldsymbol{\omega}}$ can be changed arbitrarily applying corresponding torques $\mathbf{T}$ in the joints (according to eq. (1)). By contrast, the changes in velocities $\boldsymbol{\omega}$ correspond to actual accelerations, and the changes in coordinates $\mathbf{q}$ correspond to actual velocities (double integrator behaviour). This allows the coefficients $\alpha_0, \ldots, \alpha_3$ in eq. (7) at each moment to be considered constants and the accelerations $\dot{\boldsymbol{\omega}}$ 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 $\dot{\boldsymbol{\omega}}$ 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}.

Similar consideration can be given for the x-coordinate of CM, given by eq. (5). After dual differentiation of this equation we get the formula for the acceleration a_{cm} of x_{cm}:

$$a_{cm} = \beta_1 \dot{\omega}_1 + \beta_2 \dot{\omega}_2 + \beta_3 \dot{\omega}_3 \tag{8}$$

where $\beta_1, \ldots, \beta_3$ are non-linear functions in $\mathbf{q}$ and $\boldsymbol{\omega}$. Due to the same argumentation as in the case of the ZMP-plane, at each moment and for each given a_{cm} eq. (8) describes a plane in the acceleration space. This plane is called the *CM-plane*. Again, by choosing the actual acceleration of the system $\dot{\boldsymbol{\omega}}$ from the CM-plane, the biped is forced to move in the manner, that x_{cm} is changing with the acceleration equal to a_{cm}.

To achieve the movement by which the x-coordinate of CM accelerates with a_{cm} and the x-coordinate of the ZMP remains simultaneously equal to x_{zmp}, the acceleration vector $\dot{\boldsymbol{\omega}}$ should belong to the line in acceleration space constituted by the intersection of two planes, defined with eq. (7) and (8). This line will be denoted as *CM-ZMP-line*.

The block *non-linear ctrl* in Fig. 3 is implemented as an algorithm, which changes the original acceleration vector $\dot{\boldsymbol{\omega}}$ by projecting it onto the CM-plane and CM-ZMP-line:

1. $a_{cm} = k_x \left(x_{cm} - x_{cm}^{actual}\right) + k_v \left(v_{cm} - v_{cm}^{actual}\right)$
2. $\dot{\boldsymbol{\omega}}^* = projection_{CM-plane}(\dot{\boldsymbol{\omega}})$
3. calculate x-coordinate of ZMP, x_{zmp}^*, corresponding to $\dot{\boldsymbol{\omega}}^*$
4. if($x_{zmp}^* < x_{min}$ or $x_{zmp}^* > x_{max}$)
   $x_{zmp}^* = x_{min}$ or $x_{zmp}^* = x_{max}$
   $\dot{\boldsymbol{\omega}}^* = projection_{CM-ZMP-line}(\dot{\boldsymbol{\omega}})$
   end if

in step 1, the acceleration a_{cm} is calculated as an output of the PD controller which steers the actual x-coordinate of

CM x_{cm}^{actual} to the desired position x_{cm}, given by eq. (6), and at the desired velocity υ_{cm}, which is equal to 0 in the presented examples. υ_{cm}^{actual} is the actual velocity of x_{cm}, k_x and k_υ are coefficients of the PD controller. In step 2, the initial acceleration vector $\dot{\boldsymbol{\omega}}$ is projected onto the CM-plane, described by eq. (8). The resulting acceleration vector $\dot{\boldsymbol{\omega}}^*$ ensures that x_{cm}^{actual} moves with a_{cm} as computed in step 1. In step 3, the new x-coordinate of ZMP, x_{zmp}^*, which corresponds to the $\dot{\boldsymbol{\omega}}^*$ from step 2, is calculated with eq. (4). After that, in step 4, the stability condition (3) is checked. If this condition is violated, x_{zmp}^* has to be set to the margin value which was violated. After setting x_{zmp}^*, the CM-ZMP-line is uniquely defined and the initial acceleration vector $\dot{\boldsymbol{\omega}}$ can be projected onto this line. Note, that the last projection compels the x_{cm}^{actual} to move in the same way as the projection in step 2, due to the fact that the CM-ZMP-line lies in the CM-plane. Additionally, the projection in step 4 ensures the satisfaction of stability condition (3).

The operator $projection_*(*)$ in steps 2 and 4 can be implemented in different ways, important is that the resulting acceleration vector $\dot{\boldsymbol{\omega}}^*$ lies on the corresponding plane or line. Three different definitions of this operator (see Fig. 4) were investigated. For the sake of clarity in Fig. 4 only two of

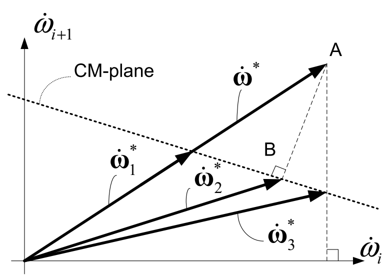

Fig. 4. Three definitions for operator $projection_*(*)$.

three dimensions of the acceleration space are shown. In the first definition the length of the origin vector $\dot{\boldsymbol{\omega}}$ is changed in such a way that its tip lies in the CM-plane. The resulting vector is denoted as $\dot{\boldsymbol{\omega}}_{\mathbf{1}}^*$. In the second definition, the end of the resulting vector, denoted as $\dot{\boldsymbol{\omega}}_{\mathbf{2}}^*$, is given by point B, which is the crossing point of the perpendicular from the end of origin vector $\dot{\boldsymbol{\omega}}$ (point A) to the CM-plane. In the third definition, in case of projection onto the CM-plane two coordinates of the origin vector $\dot{\boldsymbol{\omega}}$ remain unchanged and only one is adjusted, whereas in case of the projection onto the CM-ZMP-line one coordinate remains unchanged and two others are adjusted. The resulting vector is denoted as $\dot{\boldsymbol{\omega}}_{\mathbf{3}}^*$. The simulation of different movements has shown that the second definition of the operator $projection_*(*)$ leads to the best movement performance (see sec. III-C). The implementation of each of these three operators requires only the solution of a corresponding system of linear equations with dimension less than the dimension of the vector $\boldsymbol{\omega}^*$. This allows the usage of more complicated 3D models for biped description as presented here.

C. The Simulation Results

The following parameters for the biped model are used in the simulation: the mass of the thigh and shank was set to $5\,kg$ each; the mass of the trunk to $40\,kg$; the length of the shank and thigh was set to $0.5\,m$ and the length of the trunk junk to $1\,m$. All body parts were modelled as solid cylinders with uniform mass distribution. It was assumed that the distance between the ankle joint in the foot and the floor is negligible and that there is no slippage between the foot and the floor.

In blocks $ctrl\,q_1$, $ctrl\,q_2$, $ctrl\,q_3$ (see sec. III-A and Fig. 3) PD-controllers with coefficients $k_q = 100$ and $k_\omega = 20$ were used. The coefficients k_x and k_υ in step 1 of the algorithm in sec. III-B were set to 200 and 28. In all PD-controllers the damping ξ was set equal to 1, therefore the coefficients satisfy the following relation:

$$k_{\omega,\upsilon} = 2\xi\sqrt{k_{q,x}}$$

These coefficients have an impact on the duration of the movement and on the required maximal torques. The stability of the movement in not affected by the values of these coefficients in a wide range.

The computation in the control loop runs at $1\,kHz$. MATLAB/SIMULINK was used as simulation environment.

The biped had to perform deep sitting down from initial pelvis position $x_{pelvis} = 0\,m$, $y_{pelvis} = 1.0\,m$, $q_3 = 0^\circ$ to the end position $x_{pelvis} = -0.35\,m$, $y_{pelvis} = 0.4\,m$. After that, the standing up to the initial pelvis position was performed.

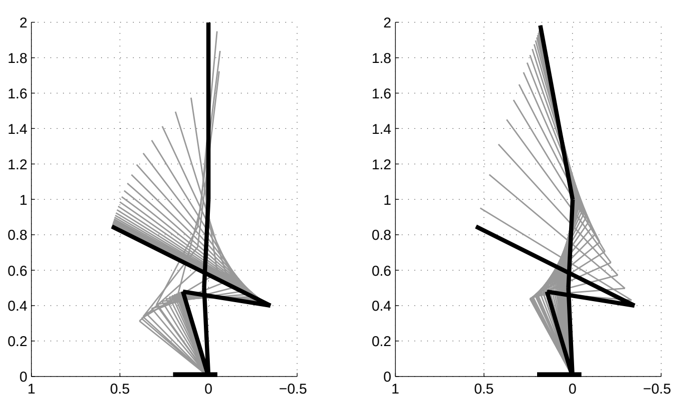

Fig. 5. Resulting movement for sitting down (left) and standing up (right).

The diagrams in Fig. 5 illustrate the resulting movement. Fig. 6-10 reveal the details of the movement. In the first $2\,s$ the biped performs sitting down after that standing up is performed. Fig. 6 shows the time dependent behavior of the pelvis coordinates and of the absolute trunk orientation $q_{3abs} = q_1 + q_2 + q_3$. Fig. 7 shows the error reduction for these variables. The trajectories for the x-coordinates of ZMP and CM as well as the velocity of the x-coordinate of CM (normalized to 0.1) are shown in Fig. 8. The x-coordinate of ZMP x_{zmp} remains within the chosen range $[-0.05\,m;\,0.2\,m]$ during the whole movement. The x-coordinate of CM x_{cm} goes to the given value $0.075\,m = (-0.05 + 0.2)/2\,m$ at

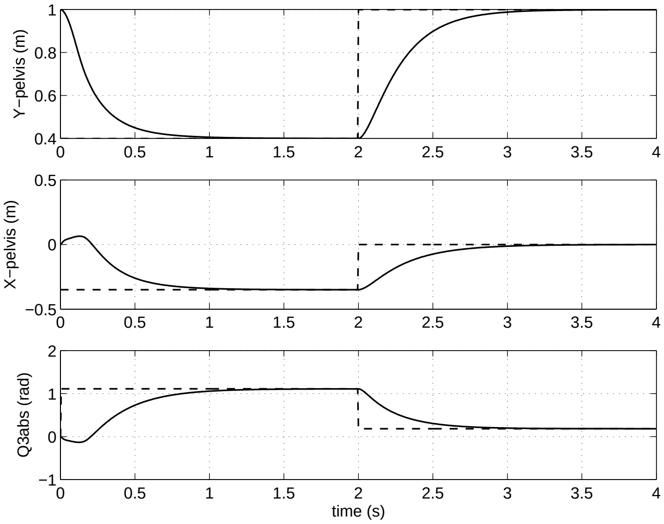

Fig. 6. The coordinates of the biped reference point and the trunk orientation.

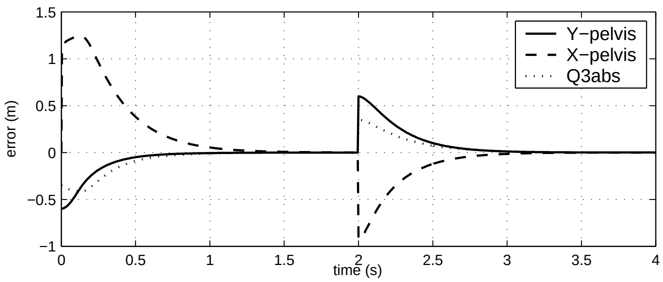

Fig. 7. Error reduction for the coordinates of the reference point and trunk orientation.

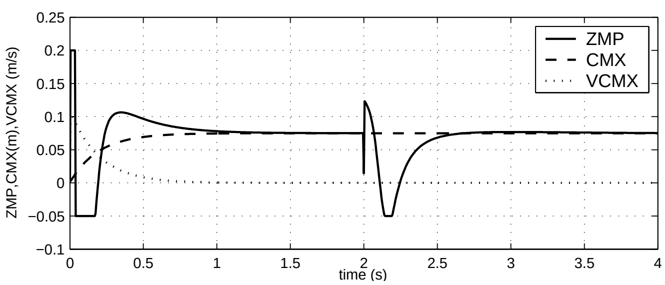

Fig. 8. Trajectories for the ZMP, CM and velocity of CM.

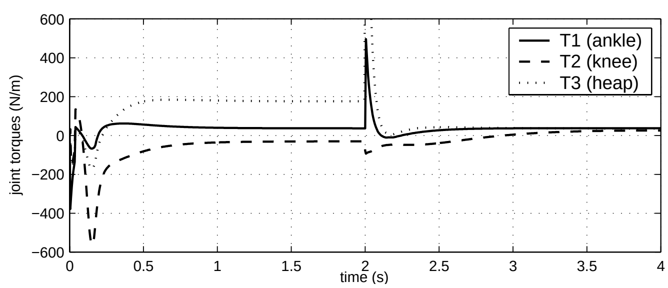

Fig. 9. Torques in joints.

the end of sitting down and remains in this position during standing up.

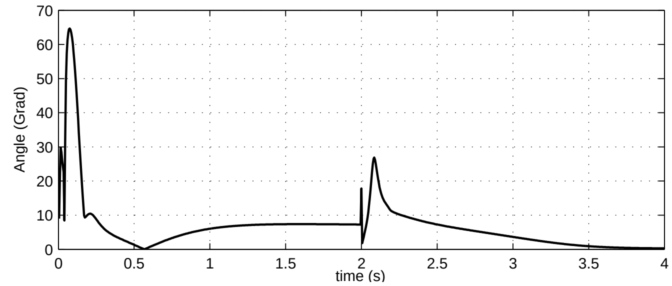

Fig. 10. The angle between the initial acceleration vector $\dot{\boldsymbol{\omega}}$ and its projection $\dot{\boldsymbol{\omega}}^*$.

Fig. 9 shows the applied torques and Fig. 10 the angle between the initial acceleration vector $\dot{\boldsymbol{\omega}}$ and its projection $\dot{\boldsymbol{\omega}}^*$, computed with the presented algorithm. Intuitively, it is clear that the projection onto the line imposes more restrictions than the projection onto the plane. This is justified in Fig. 10: The difference in the direction of both vectors rises when the operator $projection_{CM-ZMP-line}(*)$ in step 4 of the algorithm is active and x_{zmp} is equal to the upper or to the lower limit.

The authors would like to underline the importance of the operator $projection_{CM}(*)$ in the control algorithm. The steering of the reference point to the goal position is achieved by the outer control loop (see Fig. 3 and sec. III-A) and the stability condition (3) will be satisfied alone by projection of the initial acceleration vector $\dot{\boldsymbol{\omega}}$ onto the ZMP-plane. So it could be asked whether the proposed algorithm could be simplified by using only one operator for projection onto the ZMP-plane, but that is impossible. The proposed method is a local one and, despite the instantaneous satisfaction of the stability condition (3), the biped could be steered into a singular region, where the stability could be sustained only by steady speed increasing of the moving parts. In this case, the torque limits and/or ranges for possible movement in actuators are reached very quickly and the system collapses. This is

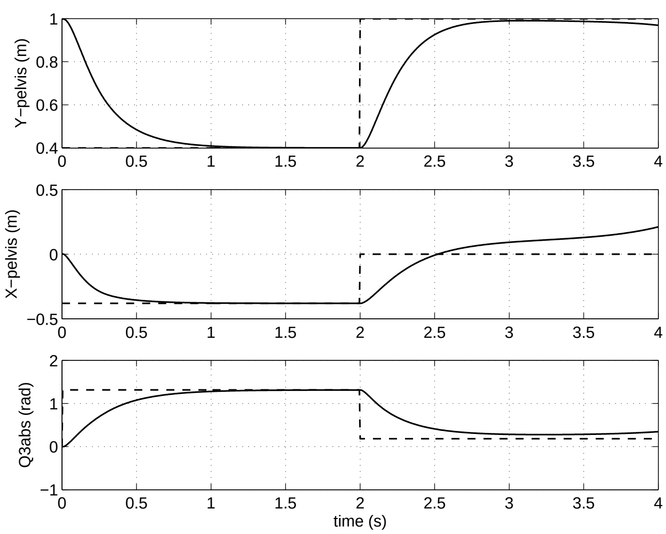

Fig. 11. The coordinates of the biped reference point and the trunk orientation: modified algorithm.

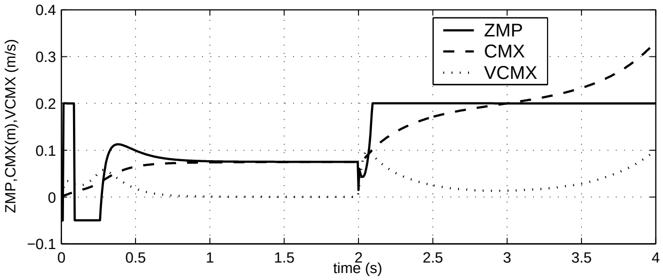

Fig. 12. Trajectories for the ZMP, CM and velocity of CM: modified algorithm.

illustrated in Fig. 11 and 12. In this trial the biped had to perform the same movement as specified above. Instead of using control algorithm in sec. III-B, the initial acceleration vector $\dot{\boldsymbol{\omega}}$ was projected onto the ZMP-plane according to step 4. As shown in Fig. 11 and 12 sitting down was performed successfully, but while standing up the system was steered into a singular region. After approx. $2.5\,s$ the x-coordinate of the pelvis increases steadily (see Fig. 11). The biped tilts forward with increasing speed and is going to tip over the foot edge after the torques in the actuators are not able to maintain the stability by increasing inertial forces of the moving parts. Fig. 12 shows that the CM has left the foot print and moves to the left with increasing speed. In this simulation the second definition of the operator $project_{ZMP}(*)$ was used (see sec. III-B). With two other definitions even sitting down was not possible.

On the other side the stable movement can not be provided only by the operator $projection_{CM}(*)$. As shown in Fig. 8 and 12, due to the inertial forces, the x-coordinate of the ZMP reaches the stability margins already at the beginning of the movement and the robot would have fallen down without considering the ZMP-plane in step 4 of the algorithm.

IV. Algorithm Stability

By algorithm stability the authors mean the stability of the movement, as defined in sec. II-B, and asymptotically achieving the movement goal.

The stability of the movement is given by the algorithm construction: acceleration vector $\dot{\boldsymbol{\omega}}^*$ applied to the system ensures the satisfaction of stability condition (3).

The x-coordiante of CM has the behaviour of a PD-controlled double integrator. If the initial position of the x-coordiante of CM belongs to $[x_{min}; x_{max}]$, the end position is given by eq. (6) and the damping is equal to 1, the x-coordiante of CM remains in $[x_{min}; x_{max}]$ during the whole movement. It ensures that the singular regions mentioned in sec. III-C will be avoided. The proof of this statement as well as the strong definition of singular regions shall not be presented here. In the case of mevements like walking or running the x-coordiante of CM could be outside of $[x_{min}; x_{max}]$ and its end velocity could be different from 0.

If step 4 in the algorithm is not active beginning from some arbitrary time point, the biped moves asymptotically towards its goal. This can be shown easily for the third definition of the operator $projection_{CM}(*)$. Two accelerations from three, e.g. $\dot{\omega}_1$ and $\dot{\omega}_2$, are not affected by $projection_{CM}(*)$ and therefore the corresponding generalized coordinates q_1 and q_2 reach the goal asymptotically controlled by $ctrl\,q_1$ and $ctrl\,q_2$. Due to the above described control of the x-coordinate of CM, for an arbitrary chosen precision ε there exists a time point from which eq. (5) is satisfied whith ε. This means that the changing of vector $\mathbf{q}$ is restricted to the two dimensional manifold, therefore, q_3 reaches the goal asymptotically together with q_1 and q_2. The case of the active step 4 is much more complicated and is not considered here.

V. Conclusion

It was shown that the trajectory generation for stable movement of a biped and the control along these trajectories could be merged into a single algorithm in form of a feedback controller. The movements of a biped in a sagittal plane such as sitting down and standing up were considered. The performance of the algorithm was demonstrated in a simulation. In the simulation it was also verified that the movement stability is not sensitive to the parameters of the algorithm. As mentioned above, the application field for the algorithm ranges from global control to online trajectories computation.

The presented version of the algorithm does not consider the limitation of the torques in actuators, which is very important for practical application. The authors are working on this problem. The adaptation of the coefficients in the controllers (decreasing at the beginning of the movement and increasing at the end), is one possible approach.

The application of the algorithm for dealing with movements like walking can be found in [11].

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*, 1998, pp. 1321–1326.

[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*, 2001, pp. 4140–4145.

[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. Springer, 1996, pp. 285–292.

[5] S. Kudoh, T. Komura, and K. Ikeuchi, "The dynamic postural adjustment with the quadratic programming method," in *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002, pp. 2563–2568.

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

[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*, 2001, pp. 237–244.

[8] M. Vukobratović, "On the stability of biped locomotion," *IEEE Trans. on Biomedical Engineering*, vol. 17, no. 1, pp. 25–36, 1970.

[9] A. Goswami, "Foot rotation indicator (FRI) point: A new gait planning tool to evaluate postural stability of biped robots," in *Proceedings of the IEEE Int. Conf. on Robotics & Automation*, 1999, pp. 47–52.

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

[11] K. Kondak and G. Hommel, "Control algorithm for stable walking of biped robots," in *Proceedings of the Sixth Int. Conf. on Climbing and Walking Robots (CLAWAR)*, 2003, accepted for publication.