HAL
open science

Stabilization of the Capture Point Dynamics for Bipedal Walking based on Model Predictive Control

Manuel Krause, Johannes Englsberger, Pierre-Brice Wieber, Christian Ott

To cite this version:

Manuel Krause, Johannes Englsberger, Pierre-Brice Wieber, Christian Ott. Stabilization of the Capture Point Dynamics for Bipedal Walking based on Model Predictive Control. IFAC Symposium on Robot Control, Sep 2012, Dubrovnik, Croatia. pp.165-171, 10.3182/20120905-3-HR-2030.00165 . hal-02487644

HAL Id: hal-02487644

https://inria.hal.science/hal-02487644

Submitted on 21 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Stabilization of the Capture Point Dynamics for Bipedal Walking based on Model Predictive Control

Manuel Krause * **Johannes Englsberger** *
Pierre-Brice Wieber ** **Christian Ott** *

* *German Aerospace Center (DLR), D-82234 Wessling, Germany (email: christian.ott@dlr.de)*
** *INRIA, 38334 St. Ismier Cedex, France*

Abstract: Considering a reduced center-of-mass model for bipedal walking, by utilizing the Capture Point as a system coordinate one can separate the stable and unstable components of the dynamics. In this paper, previous works on the stabilization of the Capture Point dynamics are extended by employing model predictive control (MPC). This allows to explicitly incorporate constraints on the zero-moment-point (ZMP) in the controller design. The proposed Capture-Point-MPC is evaluated in simulation and experiments with the DLR-Biped.

Keywords: Bipedal walking, Capture Point Dynamics, Model Predictive Control, Application

1. INTRODUCTION

Bipedal walking is one of the core skills for humanoid robots and as such received considerable attention in the robot control literature. The problem has been tackled by control of simplified dynamical models (Kajita et al. (2003); Choi et al. (2007)), planning based approaches for the complete multi-body dynamics (Suleiman et al. (2008)), as well as by utilizing physical embodiment as in passive dynamic walkers (McGeer (1990)).

From a modeling point of view, bipedal walking systems are in general best described by floating base systems, in which the contact between the feet and the ground introduces additional constraints during single and double support. In this way, the rigorous modeling leads to underactuated mechanical systems. Vukobratovic and Stepanenko (1972) introduced the zero moment point (ZMP) as a representation of the horizontal moments of the ground reaction force. By preventing the ZMP to reach the borders of the support polygon it is possible to ensure that the feet stay in firm contact with the ground. In this way, the underactuation problem can be avoided.

The linear inverted pendulum model for bipedal walking was introduced in (Kajita et al. (2001)) as a simplified model for control and is since then widely used for generation of walking patterns (Sugihara et al. (2002); Choi et al. (2007)). Kajita et al. (2003) used an extension of LQR control by preview action for a closed loop approach to bipedal walking pattern generation. This approach was extended by Wieber (2006) based on a model predictive control (MPC) formulation, which allows to exactly consider the constraints on the ZMP. Further extensions allowed to handle more general problems including variable foot placement in (Diedam et al. (2008)). Moreover, MPC was applied to push recovery control in (Stephens (2011)). Pratt et al. (2006) introduced the 'Capture Point' (CP) as the point on the floor, where the robot should place its ZMP in order to asymptotically stop the motion of the center of mass (COM). In (Pratt et al. (2006)), the CP was used as a tool for analyzing the push recovery problem. The same concept was introduced by Hof (2008) in the biomechanics literature under the notation 'extrapolated center of mass'. Moreover, it can be shown that the CP also corresponds to the 'divergent component of motion' of the linear inverted pendulum dynamics, which was utilized in (Takenaka et al. (2009)) for realtime generation of walking patterns. Based on the dynamics of the capture point Englsberger et al. (2011) proposed a feedback tracking controller in which the constraints on the ZMP were added by an additional projection of the control input.

In this paper, we propose to utilize model predictive control to stabilize the unstable dynamics related to the CP. Thereby, we aim at a feedback control solution rather than a walking pattern generator. Therefore, we consider the ZMP as the control input and combine the outer loop MPC with an inner loop ZMP controller. In contrast to the controller presented in (Englsberger et al. (2011)), the utilization of model predictive control allows the incorporation of ZMP constraints in the controller design.

The paper is organized as follows. Section 2 reviews the concentrated mass model for bipedal walking and its use for walking pattern generation. In Sec. 3 we discuss the capture point dynamics and formulate an MPC for feedback stabilization. Simulations and experiments are presented in Sec. 4. Finally, Sec. 5 concludes the paper.

2. BACKGROUND

2.1 Concentrated mass model for bipedal walking

Bipedal walking in general is an underactuated control problem, whenever the foot happens to loose planar con-

tact with the ground. The ZMP (Vukobratovic and Stepanenko (1972)) was introduced as a concise representation of the contact force and from its very definition it follows that such underactuated situations can be avoided if the controller ensures that the zero moment point stays strictly within the support polygon. The location of the ZMP $(p_x, p_y, 0)$ can be computed based on the position (c_x, c_y, c_z) of the total center of mass and the torque τ_x, τ_y, τ_z corresponding to the change of the system's total angular momentum around the COM:

$$p_x = \frac{c_x(\ddot{c}_z + g) - c_z\ddot{c}_x}{\ddot{c}_z + g} - \frac{\tau_y}{m(\ddot{c}_z + g)} \ , \tag{1}$$

$$p_y = \frac{c_y(\ddot{c}_z + g) - c_z\ddot{c}_y}{\ddot{c}_z + g} + \frac{\tau_x}{m(\ddot{c}_z + g)} \ , \tag{2}$$

wherein m and g denote the total mass and the gravity acceleration. If the dynamical effects of the vertical motion

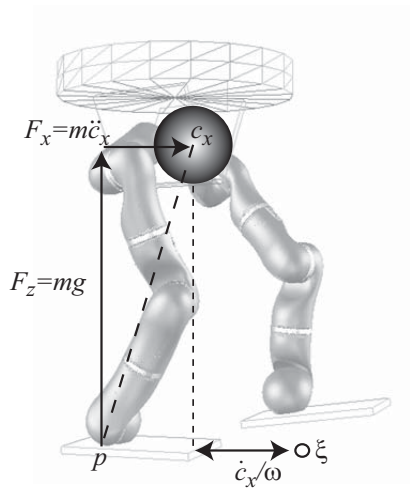

Fig. 1. Capture Point

and the change of angular momentum can be neglected, (1)-(2) simplifies to the well-known equations of motion of the linear inverted pendulum model (Kajita et al. (2001))

$$\ddot{\boldsymbol{x}} = \omega^2(\boldsymbol{x} - \boldsymbol{p}) \ , \tag{3}$$

with the horizontal coordinates of the COM $\boldsymbol{x} = (c_x, c_y)$ and ZMP $\boldsymbol{p} = (p_x, p_y)$, and the parameter $\omega = \sqrt{g/c_z}$. Despite its simplicity and its known limitations, (3) describes the macroscopic dynamics of bipedal walking surprisingly well and has successfully been used for the generation of walking trajectories of several position controlled humanoid robots (Kajita et al. (2003); Wieber (2006)).

One widely used approach is the combination of a walking pattern generation algorithm with an underlying ZMP based feedback stabilizer. From a control point of view, one challenge of both the walking pattern generation and the feedback control are the requirement that the ZMP should not reach the borders of the support polygon.

2.2 MPC for Walking Pattern Generation

For the purpose of trajectory generation it is useful to consider the motion of the center of mass as the manipulated quantity and use it to steer the ZMP along the consecutive stance foot positions. Kajita et al. (2003) therefore introduced the jerk of the COM motion as an input $\boldsymbol{u} = \dddot{\boldsymbol{x}}$ to (3) in order to obtain the ZMP as a system output $\boldsymbol{y} = \boldsymbol{p}$. The resulting model is given by the linear system

$$\dddot{\boldsymbol{x}} = \boldsymbol{u} \ , \tag{4}$$

$$\boldsymbol{y} = \boldsymbol{x} - \omega^2\ddot{\boldsymbol{x}} \ , \tag{5}$$

which was controlled by a discrete LQR controller with preview action in (Kajita et al. (2003)). Wieber (2006) proposed to apply linear model predictive control to the discrete time implementation of the model (4)-(5) and used the resulting COM and ZMP trajectories as desired trajectories for an underlying feedback stabilizer. Thereby, the MPC framework allows to ensure the fulfillment of the support polygon constraints in the generated ZMP trajectories. The approach was extended and generalized in (Diedam et al. (2008)) to adaptive foot placement.

3. CAPTURE POINT MODEL PREDICTIVE CONTROL

The model (4)-(5) is well suited for trajectory generation but is problematic for the design of feedback controllers, since it contains the acceleration $\ddot{\boldsymbol{x}}$ as an additional virtual state variable in addition to the physical state variables $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$. A state feedback controller based on this model thus can lead to an algebraic feedback loop causing an ill-posed feedback system.

In this paper we aim instead at a feedback control law for the model (3). Therefore, we consider the ZMP (which is related to the contact forces) rather than the COM motion as a control input.

3.1 Capture Point Dynamics

In (Pratt et al. (2006)) the 'Capture Point' (CP) has been introduced as the specific (constant) location for the ZMP, which asymptotically brings the COM motion to a rest. As shown in (Pratt et al. (2006)) and (Hof (2008)), this point can be determined by computing the solution of the linear differential equation (3) for a constant ZMP $\boldsymbol{p}$ and requiring that the COM converges to $\boldsymbol{p}$. This leads to the definition of the CP

$$\boldsymbol{\xi} := \boldsymbol{x} + \frac{1}{\omega}\dot{\boldsymbol{x}} \ . \tag{6}$$

Apart from the physical interpretation of $\boldsymbol{\xi}$ as the 'Capture Point', which allows to asymptotically stop the robot motion, we can see that $\boldsymbol{\xi}$ represents a composite variable of a position dependent and a velocity dependent term. Such composite variables are well know from the literature of adaptive control (see e.g. Slotine and Li (1991)).

Considering $\boldsymbol{\xi}$ as a state variable, we can perform a state transformation for the system (3) from the state $(\boldsymbol{x}, \dot{\boldsymbol{x}})$ into the new state variables $(\boldsymbol{x}, \boldsymbol{\xi})$. This leads to

$$\dot{\boldsymbol{x}} + \omega\boldsymbol{x} = \omega\boldsymbol{\xi} \ , \tag{7}$$

$$\dot{\boldsymbol{\xi}} - \omega\boldsymbol{\xi} = -\omega\boldsymbol{p} \ . \tag{8}$$

We can see that in the new coordinates the system has a cascaded form. The capture point dynamics (8) represents

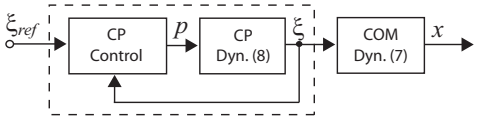

Fig. 2. Capture Point Control for stabilizing the unstable part of the cascaded system

an unstable first order system (for each of the directions x and y), which drives the first order COM dynamics (7). Notice that this system structure is a direct consequence of utilizing the composite variable $\boldsymbol{\xi}$ as a state variable.

3.2 Controller design

In (Englsberger et al. (2011)) two different linear feedback controllers were proposed for the model (8). Both controllers had the form

$$\boldsymbol{p} = k_d \boldsymbol{\xi}_d + (1 - k_d)\boldsymbol{\xi} \ , \tag{9}$$

where the negative gain $k_d < 0$ was related to the open loop solution of (8) and was chosen either time-varying or constant in the two controllers. As a desired trajectory, a time shifted reference trajectory was used, i.e. $\xi_d(t) = \xi_{ref}(t + T_0)$, and this time shift parameter T_0 also affected the design of k_d. The main idea behind the control law (9) was to focus on the stabilization of the unstable part of the dynamics and utilize the particular cascaded system structure of (7)-(8) for analyzing the stability of the whole system (see Fig. 2). Recently, an extension of this control law to the more general nonlinear model (1)-(2) was proposed in (Englsberger and Ott (2012)).

In order to respect the constraints on the ZMP, a projection onto the support polygon was proposed in (Englsberger et al. (2011)), but the stability analysis was performed for the unconstrained control law. This limitation of the control law from Englsberger et al. (2011) motivates the approach in this paper in which the ZMP constraint will be incorporated in an MPC based feedback controller for the capture point dynamics (8) (see Fig. 3).

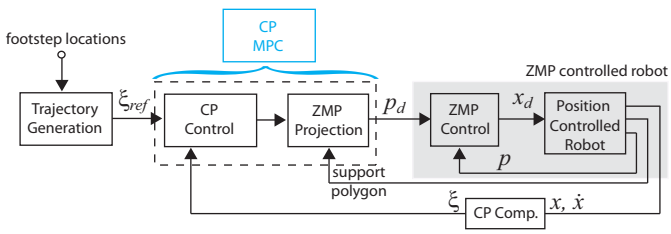

Fig. 3. Capture Point Model Predictive Control (CPMPC) compared to the control approach from Englsberger et al. (2011) where the feedback controller was designed independently from the ZMP constraint.

Similarly as in (9), we are interested in stabilizing only the unstable part of the dynamics. Therefore, we use (8) as a prediction model and we consider $\boldsymbol{p}$ as the control input, which has to be realized by an inner loop force or ZMP controller. Considering a discrete time implementation with piecewise constant control inputs for $\boldsymbol{p}$, we obtain (from (8)) the prediction model

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{A}\boldsymbol{\xi}_k + \boldsymbol{B}\boldsymbol{p}_k \quad , \tag{10}$$

with

$$\boldsymbol{A} = \begin{bmatrix} e^{\omega T} & 0 \\ 0 & e^{\omega T} \end{bmatrix} \qquad \boldsymbol{B} = \begin{pmatrix} 1 - e^{\omega T} \\ 1 - e^{\omega T} \end{pmatrix} \ , \tag{11}$$

where T is the sampling time of the prediction model. Based on the current state $\boldsymbol{\xi}_k$ and the future control inputs $\boldsymbol{p}_k$ the prediction of (10) j time steps ahead is simply given by $\boldsymbol{\xi}_{k+j} = \boldsymbol{A}^j \boldsymbol{\xi}_k + \sum_{i=0}^{j-1} \boldsymbol{A}^{j-1-i} \boldsymbol{B} \boldsymbol{p}_{k+i}$ and we can summarize the predicted CP for the next N steps in the vector

$$\boldsymbol{\Xi} = \boldsymbol{F}_\xi \boldsymbol{\xi}_k + \boldsymbol{F}_p \boldsymbol{P} \ , \tag{12}$$

with

$$\boldsymbol{F}_\xi = \begin{bmatrix} \boldsymbol{A} \\ \vdots \\ \boldsymbol{A}^N \end{bmatrix} \quad \boldsymbol{F}_p = \begin{bmatrix} \boldsymbol{A}^0 \boldsymbol{B} & \cdots & \boldsymbol{0} \\ \cdots & \ddots & \vdots \\ \boldsymbol{A}^{N-1} \boldsymbol{B} & \cdots & \boldsymbol{A}^0 \boldsymbol{B} \end{bmatrix} \ , \tag{13}$$

and the predicted capture point values and the future values for the ZMP included in the vectors

$$\boldsymbol{\Xi} = \begin{pmatrix} \boldsymbol{\xi}_{k+1} \\ \vdots \\ \boldsymbol{\xi}_{k+N} \end{pmatrix} \ , \quad \boldsymbol{P} = \begin{pmatrix} \boldsymbol{p}_k \\ \vdots \\ \boldsymbol{p}_{k+N-1} \end{pmatrix} \ . \tag{14}$$

The considered control objective is the tracking of a given reference trajectory $\boldsymbol{\xi}_{ref}$ for the capture point, while keeping the ZMP strictly within the support polygon. As an objective function for the MPC at time instant k, we choose

$$J_k = \frac{1}{2} \sum_{i=1}^{N} q_i (\boldsymbol{\xi}_{k+i} - \boldsymbol{\xi}_{ref,k+i})^T (\boldsymbol{\xi}_{k+i} - \boldsymbol{\xi}_{ref,k+i}) + r_i \Delta \boldsymbol{p}_{k+i-1}^T \Delta \boldsymbol{p}_{k+i-1} \ , \tag{15}$$

with positive weights $q_1 \cdots q_N$ for the output and $r_1 \cdots r_N$ for the rate of change of the control input. The second term in (15) is added for obtaining a smoother control signal. Equivalently we can write this cost function in the form

$$J_k = \frac{1}{2} (\boldsymbol{\Xi} - \boldsymbol{\Xi}_{ref})^T \boldsymbol{Q} (\boldsymbol{\Xi} - \boldsymbol{\Xi}_{ref}) + \Delta \boldsymbol{P}^T \boldsymbol{R} \Delta \boldsymbol{P} \ , \tag{16}$$

with the diagonal matrices $\boldsymbol{Q} = diag(q_i)$ and $\boldsymbol{R} = diag(r_i)$ and the vector $\boldsymbol{\Xi}_{ref}$ of future reference trajectory values $\boldsymbol{\xi}_{ref,k+i}$. The rate of change of the future ZMP can be related to $\boldsymbol{P}$ via

$$\Delta \boldsymbol{P} = \boldsymbol{\Theta} \boldsymbol{P} - \boldsymbol{e}_1 \boldsymbol{p}_{k-1} \tag{17}$$

where

$$\boldsymbol{\Theta} = \begin{pmatrix} \boldsymbol{I} & 0 & \cdots & & \\ -\boldsymbol{I} & \boldsymbol{I} & 0 & \cdots & \\ 0 & -\boldsymbol{I} & \boldsymbol{I} & 0 & \cdots \\ \vdots & \cdots & \ddots & \ddots & \end{pmatrix} \ , \quad \boldsymbol{e}_1 = \begin{pmatrix} \boldsymbol{I} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \ . \tag{18}$$

By utilizing (12), we can write (16) in the standard form of a quadratic programming (QP) problem (Ferreau et al. (2008)):

$$J_k(\boldsymbol{P}) = \frac{1}{2} \boldsymbol{P}^T \boldsymbol{H} \boldsymbol{P} + \boldsymbol{g}^T \boldsymbol{P} \ , \tag{19}$$

where

$$\boldsymbol{H} = \boldsymbol{\Theta}^T \boldsymbol{R} \boldsymbol{\Theta} + \boldsymbol{F}_p^T \boldsymbol{Q} \boldsymbol{F}_p \ ,$$
$$\boldsymbol{g} = \boldsymbol{F}_p^T \boldsymbol{Q} (\boldsymbol{F}_\xi \boldsymbol{\xi}_k - \boldsymbol{\Xi}_{ref}) - \boldsymbol{\Theta}^T \boldsymbol{R} \boldsymbol{e}_1 \boldsymbol{p}_{k-1} \ .$$

The presented optimization formulation contains both x and y direction, while the representation (3) describes two decoupled system equations in lateral and sagittal direction. However, the inequality constraints in the following section contain constraints which introduce additional couplings between the two directions. As a consequence it is not possible to consider the control of the two directions separately.

According to the MPC framework, the current control input is computed by optimizing the cost $J_k(\boldsymbol{P})$ for the future control inputs and applying the first control action from $\boldsymbol{P}$ to the system, i.e.

$$\boldsymbol{p}_d = \boldsymbol{e}_1^T \boldsymbol{P}^* \ , \quad \boldsymbol{P}^* := \underset{\boldsymbol{P}}{\operatorname{argmin}} \, J_k(\boldsymbol{P}) \ . \tag{20}$$

In the solution of the optimization problem, additional constraints as formulated in the next section can be integrated into the controller computation.

3.3 Constraint formulation

In the following we assume that a set of desired footstep positions $\boldsymbol{r}_i \in \mathbb{R}^2$ and orientations $\alpha_i \in \mathbb{R}$ are provided (from a kinematic footstep planning algorithm) as well as a mapping function $s(k)$, which gives for each time index k the corresponding footstep i. Let $\boldsymbol{R}(\alpha) \in SE(2)$ be the rotation matrix corresponding to the footstep orientation. In the following it is assumed that the foot geometry has a simple rectangular shape. For a time index k corresponding to a single support phase, the constraints on the ZMP are given by

$$\boldsymbol{f}_{min} \leq \boldsymbol{R}(\alpha_{s(k)})(\boldsymbol{p}_k - \boldsymbol{r}_{s(k)}) \leq \boldsymbol{f}_{max} \ , \tag{21}$$

where $\boldsymbol{f}_{min}$ and $\boldsymbol{f}_{max}$ are the minimum and maximum values determined by the foot geometry.

In case of a time instant k corresponding to a double support phase, instead of (21) we would need to describe the complete support polygon, which leads to a larger set of inequality constraints. In the experiments reported in section 4 we used a simple approximation of the double support polygon by a rectangular box constraint.

3.4 A Remark on Stability

It is well know in the literature on MPC that stability of the closed loop system can be enforced by an additional terminal constraint on the state (Mayne et al. (2000)), which would mean in our case $\boldsymbol{\xi}_{k+N} = \boldsymbol{\xi}_{ref,k+N}$. However, this constraint can strongly affect the convergence of the optimization in case that the prediction horizon is not long enough. In the experiments reported in section 4, we therefore approximated this terminal constraint by a high terminal cost via the weight q_N.

3.5 Capture Point Reference Trajectory

In Englsberger and Ott (2012) a method for the generation of a CP reference trajectory from given footsteps was proposed. For a fixed ZMP $\boldsymbol{p}$ the solution of (8) is given by

$$\boldsymbol{\xi}(t) = \boldsymbol{p} + e^{\omega t} \, (\boldsymbol{\xi}_0 - \boldsymbol{p}) \ . \tag{22}$$

For a constant ZMP $\boldsymbol{p}$ the CP $\boldsymbol{\xi}$ moves away from the ZMP on a straight line. With the COM dynamics (7) we find that the COM follows the CP, its velocity vector $\dot{\boldsymbol{\xi}}$ always pointing towards the CP.

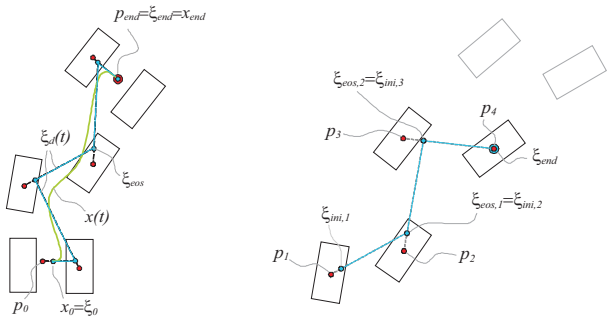

Fig. 4. Two-dimensional Capture Point manipulation: (a) Foot to foot shift (b) Preview of next three steps for CP reference

The basic idea used in this paper is to produce a walking pattern by shifting the CP during a step from one predefined footprint to the next (Fig. 4(a)). As the COM automatically follows the CP (green curve), only the CP dynamics (blue lines) has to be considered. This way, the CP and COM are shifted from the initial COM position $\boldsymbol{x}_0$ to the final COM position $\boldsymbol{x}_{end}$. The goal CP at the end of each step is denoted by $\boldsymbol{\xi}_{eos}$ ("eos" = "end of step").

The computation of $\boldsymbol{\xi}_{eos}$ is based on a backward calculation. With the final CP position $\boldsymbol{\xi}_{eos,i}$ at the end of each step and the desired ZMP position (here chosen to be in the center of the stance foot $\boldsymbol{p}f,i$) we can calculate an ideal initial CP $\boldsymbol{\xi}_{ini,i}$ for each step with $\boldsymbol{\xi}_{ini,i} = \boldsymbol{p}_{f,i} + (\boldsymbol{\xi}_{eos,i} - \boldsymbol{p}_{f,i})/(e^{\omega \, t_{step}})$, where t_{step} denotes the total time per step. This ideal initial CP $\boldsymbol{\xi}_{ini,i}$ is then used as the desired final CP position $\boldsymbol{\xi}_{eos}$ for the previous step, so $\boldsymbol{\xi}_{eos,i-1} = \boldsymbol{\xi}_{ini,i}$. In that way, from the final step (after which the robot usually comes to a stop) until the current step, all $\boldsymbol{\xi}_{eos,i}$ as well as the whole future desired trajectory of the CP (blue lines in Fig. 4(a)) can be calculated. In practice, we limit ourselves to the use of the current footprint ($\boldsymbol{p}_{f,1}$) and the three next footprints ($\boldsymbol{p}_{f,2}$, $\boldsymbol{p}_{f,3}$ and $\boldsymbol{p}_{f,4}$, see Fig. 4(b)) for the calculation of the CP tracking reference, instead of using the whole list of future footprints. This reduces the computational effort while the deviation from the trajectory generation using all future footprints is marginal.

The reference trajectory for the prediction horizon can be computed from (22) based on the footprints $\boldsymbol{p}_{f,i}$ and on $\boldsymbol{\xi}_{ini,i}$. The special feature of the proposed CP reference trajectory is that in the unperturbed case the ZMP $\boldsymbol{p}$ is always located in one of the foot centers $\boldsymbol{p}_{f,i}$, which decreases the likelihood of tilting. Holding the ZMP in the foot center during a step might be desirable whereas a discontinuity at the transition of a support phase (e.g. single to double support) can cause perturbations. Therefore, the MPC is trying to track the CP reference trajectory on the one hand

side, but to return a smooth ZMP trajectory on the other hand side.

3.6 Position based ZMP Control

In order to implement the desired ZMP $\boldsymbol{p}_d$ from (20) on a position controlled robot, an underlying ZMP control loop is required. The current ZMP can be measured by force/torque sensors in the feet of the robot. Considering the modeling assumption of the simplified dynamics (3), we can relate the desired ZMP $\boldsymbol{p}_d$ to a desired force $\boldsymbol{F}_d = m\omega^2(\boldsymbol{x} - \boldsymbol{p}_d)$ that should act on the COM. It is well known that a position based force set point regulator can be implemented by simple integral action (Roy and Whitcomb (2002)):

$$\dot{\boldsymbol{x}}_d = k_f(\boldsymbol{F}_d - \boldsymbol{F}), \tag{23}$$

with a positive force control gain k_f. Since we are interested in a ZMP controller rather than a force controller, we insert $\boldsymbol{F}_d$ and the relation between the force $\boldsymbol{F}$ and $\boldsymbol{p}$ into (23) to obtain the ZMP control law

$$\dot{\boldsymbol{x}}_d = k_f m\omega^2(\boldsymbol{p} - \boldsymbol{p}_d)\ . \tag{24}$$

4. EVALUATION

4.1 Simulation results for the nominal model

Before evaluating the results on a real robot, we analyze the performance of the model predictive controller on the nominal model (8). Figure 5 shows the capture point tracking performance for a sampling time of $40ms$ with a prediction horizon of 40 steps, i.e. $1.6s$. In the cost function, the weights for the tracking error and the change of the control input were used constant over the prediction horizon with $q = 1$ and $r = 0.1$. In Fig. 6 we see a second result with the same prediction horizon, but with a sampling time of $80ms$ and only 20 steps. Obviously, a smaller sampling time with a higher number of steps leads to a better performance for the same length of the prediction horizon.

With regard to real-time implementation, however, it is desired to keep the number of optimization parameters small. Therefore, we used a sampling period for the control policy, which was increasing over the control horizon, while we used a constant sampling period for the state prediction. Table 1 shows the sampling periods and the corresponding number of samples (i.e. optimization variables). For the state prediction, a constant sampling time of $10ms$ was used. The prediction horizon has a length of $0.89s$. Within this horizon only 9 optimization variables appear, which correspond to the control inputs related to the different control sampling times. In Fig. 7 one can see the tracking performance of this modified MPC formulation. Even if we have only 9 optimization parameters and a shorter prediction horizon, the small sampling time of $10ms$ thus leads to an improved performance compared to Fig. 5 and Fig. 6.

4.2 Experiments

The proposed application of model predictive control to the capture point dynamics with an underlying ZMP

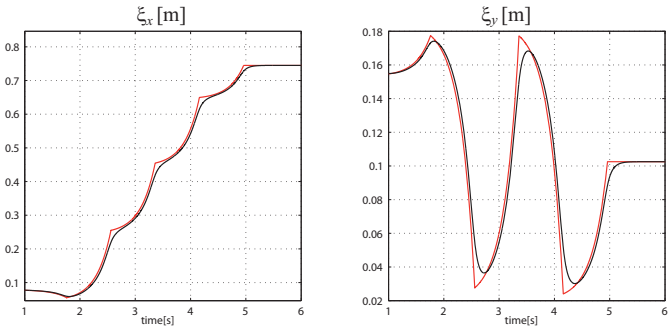

Fig. 5. Simulation of the Capture Point MPC for the nominal model with a sampling time of $40ms$ and a prediction of 40 steps. The reference trajectory is shown by the red curve.

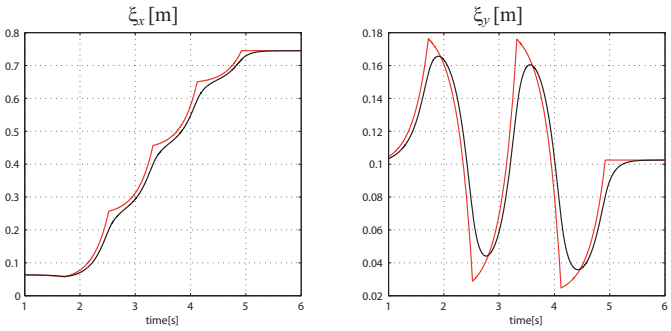

Fig. 6. Simulation of the Capture Point MPC for the nominal model with a sampling time of $80ms$ and a prediction of 20 steps. The reference trajectory is shown by the red curve.

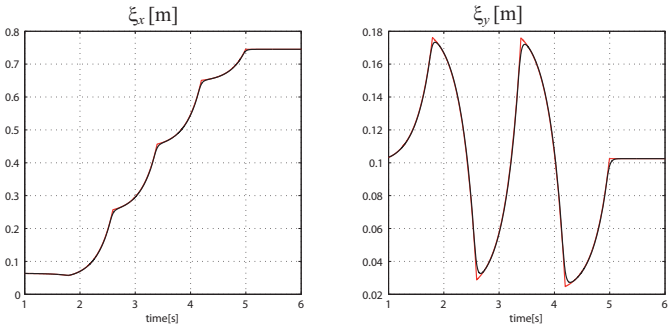

Fig. 7. Simulation of the Capture Point MPC for the nominal model with a sampling time of $10ms$ and a reduction to a set of 9 optimization parameters for each direction. The reference trajectory is shown by the red curve.

controller was implemented and evaluated for the DLR-Biped (Ott et al. (2010)) (Fig. 8). The feet of this robot have a compact size of $9.5cm$ width and $19cm$ length and are equipped with six-axis force torque sensors, which allow to determine the position of the ZMP in a sensor sampling rate of $2ms$. The realtime control and trajectory generation is implemented in a sampling time of $1ms$. The

Table 1. Increasing sampling periods for the control policy in the simulation

Control period	10ms	40ms	80ms	160ms	320ms
Control samples	1	2	4	1	1

QP problem (20) was solved with the state of the art solver from Ferreau et al. (2008) in a sampling rate of $2ms$.

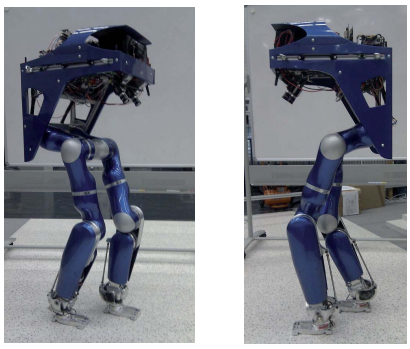

Fig. 8. DLR-Biped

The prediction horizon of the MPC was chosen as $T_h = 0.89s$ and was split into two parts for which we used different prediction sampling times T_1 and T_2. The first part corresponds to two steps with a sampling time of $T_1 = 5ms$, while in the second part of the prediction horizon, we used a sampling time of $T_2 = 10ms$. Moreover, we used an increasing sampling period for the control policy in order to reduce the size of the optimization problem (as discussed in section 4.1). Table 2 shows the distribution of the control sampling periods used in the experiments. The resulting optimization problem thus has 16 optimization variables, 8 for each direction (x and y).

Table 2. Increasing sampling periods for the control policy in the experiments

Control period	5ms	40ms	160ms	320ms
Control samples	2	2	3	1

Table 3. Optimization weights

Sampling time	T_1	T_1	T_2	T_2	T_2	T_2	T_2
Samples	1	1	8	16	16	16	32
q_i	1	1	2	4	6	8	100
r_i	0.2	0.5	10	10	10	10	10

An important aspect in the experimental evaluation was the tuning of the optimization weights along the prediction horizon, which are the controller parameters. The used optimization weights are shown in Table 3. In order to emulate the terminal constraint mentioned in section 3.4, the terminal weight was selected very high. The relatively low weights r_i at the beginning of the control horizon lead to a better disturbance rejection, but also to a larger variation of the ZMP and thus a higher sensitivity to noise.

Figure 9 shows the resulting Capture Point trajectory for a forward walk with a stride length of $0.2m$. The reference trajectory is shown in red. The corresponding ZMP is shown in Fig. 10. In order to illustrate the solution of the QP problem over time, the resulting future ZMP control inputs in x and y direction are shown in Fig. 11 and Fig. 12. One can clearly see the increasing sampling time of the control policy, which is more fine granular at the beginning of the prediction horizon and gets coarser at the end.

5. SUMMARY AND OUTLOOK

In this paper the application of the model predictive control framework to the feedback stabilization of the

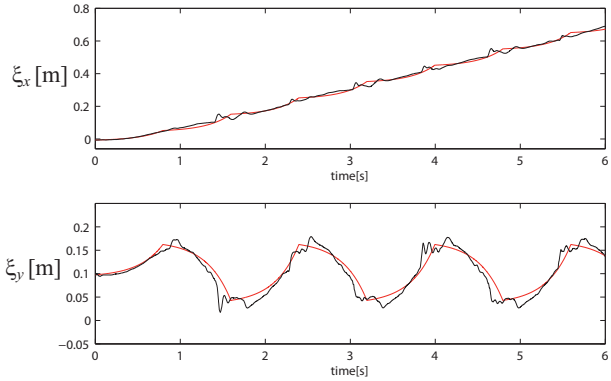

Fig. 9. Capture Point in the forward walking experiment. The reference trajectory is shown in red.

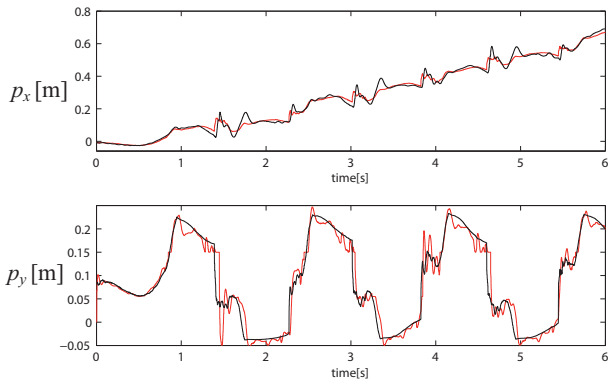

Fig. 10. ZMP in the forward walking experiment. The desired trajectory is shown in red.

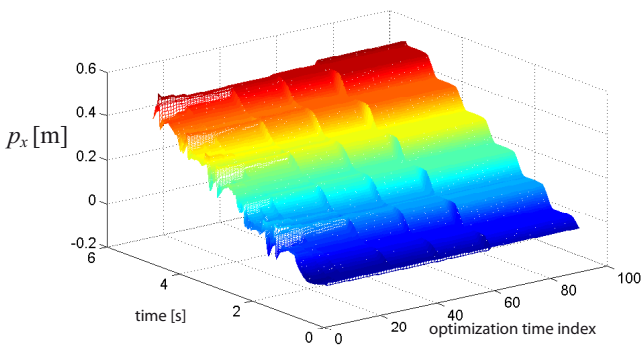

Fig. 11. Solution of the QP problem: Future ZMP control inputs in x direction. The time represents the time at which the optimization is computed, while the 'optimization time index' corresponds to the prediction time within the MPC.

capture point dynamics for bipedal walking is presented. In contrast to (Englsberger et al. (2011)), constraints on the ZMP are directly included in the controller design. The approach aims at a feedback controller, rather than a walking pattern generator. Therefore, the ZMP in the capture point dynamics was considered as the control input. The cost function in the MPC was chosen based on a given desired CP reference trajectory in combination

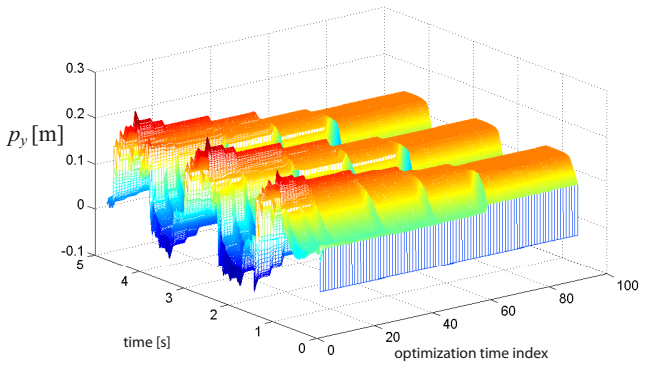

Fig. 12. Solution of the QP problem: Future ZMP control inputs in y direction. The time represents the time at which the optimization is computed, while the 'optimization time index' corresponds to the prediction time within the MPC.

with a cost for the rate of change of the control input. In the presented setup, the desired ZMP from the MPC algorithm was implemented by a position based ZMP controller. The performance of the approach was evaluated by simulations and experiments with the DLR-Biped. In order to allow for a real-time implementation with a small set of optimization variables, in the MPC a control sampling time with increasing sampling periods over the control horizon was used. The proposed algorithm achieved stable walking motions using predefined footstep locations. Our future works on this topic will focus on an elimination of the need for a pre-defined CP reference trajectory and on more efficient implementations of the optimization in the MPC algorithm.

ACKNOWLEDGEMENTS

The last author would like to thank Oliver Stasse for his comments on the formulation of the MPC problem.

This research is partly supported by the bilateral program PROCOPE under the project ID 54366426, which is financed by the German Academic Exchange Service (DAAD) with funds of the BMBF and by the French Ministry of Foreign Affairs. The second and the last author acknowledge support by the Initiative and Networking Fund of the Helmholtz Association through a Helmholtz Young Investigators Group (Grant no. VH-NG-808).

REFERENCES

Choi, Y., Kim, D., Oh, Y., and You, B.J. (2007). Posture/walking control for humanoid robot based on kinematic resolution of com jacobian with embedded motion. *IEEE Transactions on Robotics*, 23(6), 1285–1293.

Diedam, H., Dimitrov, D., Wieber, P.B., Mombaur, K., and Diehl, M. (2008). Online walking gait generation with adaptive foot positioning through linear model predictive control. In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 1121–1126.

Englsberger, J. and Ott, C. (2012). Walking stabilization for humanoid robots based on control of the capture point. *at-Automatisierungstechnik*, accepted for publication.

Englsberger, J., Ott, C., Roa, M.A., Albu-Schäffer, A., and Hirzinger, G. (2011). Bipedal walking control based on capture point dynamics. In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*.

Ferreau, H., Bock, H., and Diehl, M. (2008). An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18(8), 816–830.

Hof, A.L. (2008). The 'extrapolated center of mass' concept suggests a simple control of balance in walking. *Human Movement Science*, 27, 112–125.

Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., and Hirukawa, H. (2001). The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *IEEE Int. Conf. on Robotics and Automation*, 239–246.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *IEEE Int. Conf. on Robotics and Automation*, 1620–1626.

Mayne, D., Rawlings, J., Rao, C.V., and Scokaert, P.O.M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36, 789–814.

McGeer, T. (1990). Passive dynamic walking. *The International Journal of Robotics Research*, 9(2), 62–82.

Ott, C., Baumgärtner, C., Mayr, J., Fuchs, M., Burger, R., Lee, D., Eiberger, O., Albu-Schäffer, A., Grebenstein, M., and Hirzinger, G. (2010). Development of a biped robot with torque controlled joints. In *IEEE-RAS Int. Conf. on Humanoid Robots*, 167–173.

Pratt, J., Carff, J., Drakunov, S., and Goswami, A. (2006). Capture point: A step toward humanoid push recovery. In *IEEE-RAS Int. Conf. on Humanoid Robots*, 200–207.

Roy, J. and Whitcomb, L.L. (2002). Adaptive force control of position/velocity controlled robots: Theory and experiment. *IEEE Transactions on Robotics and Automation*, 18(2), 121–137.

Slotine, J.J.E. and Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall.

Stephens, B. (2011). *Push Recovery Control for Force-Controlled Humanoid Robots*. Ph.D. thesis, Carnegy Mellon University.

Sugihara, T., Nakamura, Y., and Inoue, H. (2002). Real-time humanoid motion generation through zmp manipulation based on inverted pendulum control. In *IEEE Int. Conf. on Robotics and Automation*, 1404–1409.

Suleiman, W., Yoshida, E., Kanehiro, F., Laumond, J.P., and Monin, A. (2008). On human motion imitation by humanoid robot. In *ICRA*, 2697–2704.

Takenaka, T., Matsumoto, T., and Yoshiike, T. (2009). Real time motion generation and control for biped robot -1st report: Walking gait pattern generation. In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 1084–1091.

Vukobratovic, M. and Stepanenko, Y. (1972). On the stability of anthropomorphic systems. *Mathematical Biosciences*, 15, 1–37.

Wieber, P.B. (2006). Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *IEEE-RAS Int. Conf. on Humanoid Robots*.