

Model-Based Predictive Bipedal Walking Stabilization

Robert Wittmann, Arne-Christoph Hildebrandt, Daniel Wahrmann, Felix Sygulla, Daniel Rixen
and Thomas Buschmann¹

Abstract—A well known strategy in bipedal locomotion to prevent falling in the presence of large disturbances is to modify drastically future motion. This is an important capability of a walking control system in order to bring humanoid robots from controlled laboratory conditions to real environment situations. This paper presents a predictive stabilization method which modifies planned center of mass and foot trajectories depending on the current state of the robot. It uses a nonlinear prediction model [1] and applies a conjugate gradient method to solve the resulting optimization problem in real-time. Furthermore, the method is integrated in the walking control system of our bipedal robot LOLA. Simulation results demonstrate the effectiveness and the advantages of the proposed method.

I. INTRODUCTION

There are several critical tasks that have to be solved if humanoid robots should operate in arbitrary unknown environments. In order to avoid falling and colliding with the environment, the walking system has to be flexible and robust. This means that it can quickly adapt to new and unknown situations. Therefore it has to first perceive the environment, detect obstacles and surfaces and adapt its motion in order to avoid collisions [2], [3]. Second, it has to be able to react to unknown disturbances which result from perception errors, unknown external forces or undetected environment properties (e.g. soft ground). For large disturbances this often requires a method to perform not only local modifications but also an adaptation of future motion. This paper deals with the second task and presents a method to adapt the robot's ideal planned motion according to sensor data. In addition to our previously published method [4] which modifies swing foot trajectories we propose a method for modifying the horizontal motion (x and y , cf. frame of reference in Fig. 1) of the center of mass (CoM) in this paper.

The paper is organized as follows: Section II gives an overview of related work. A description of the problem and the walking control system of our biped is given in Section III. The walking stabilization algorithm is described in Section IV. Section V presents first results obtained from the proposed method and Section VI concludes the paper.

II. RELATED WORK

There are many different methods for bipedal walking stabilization. This section starts with two control frameworks that include local modifications as well as an adaptation of future motion. A feedback control framework for the biped HRP2 is presented in [5], [6]. It is based on a continuous recalculation of the walking pattern with a moderate frequency

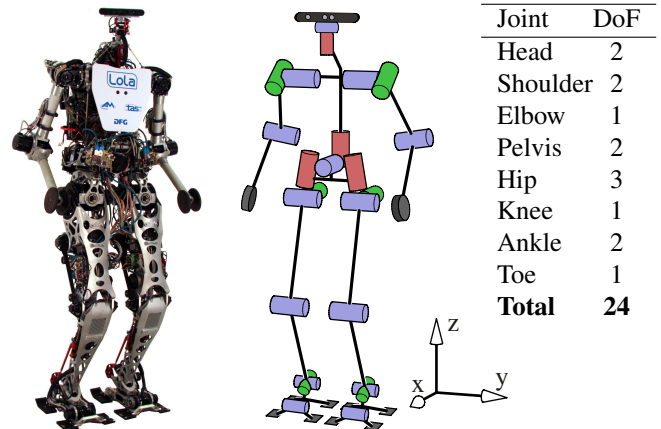


Fig. 1: Photograph and kinematic structure of LOLA.

and local adaption of the trajectories with a high frequency. The current state is used as initial value for the trajectory planning problem which is formulated as a preview control of the linear inverted pendulum model (LIPM). Ref. [7], [8] present a stabilizer of the humanoid robot ASIMO. The main feedback variable is the absolute inclination of the upper body which is treated as horizontal displacement error of the CoM and is used to calculate a reaction moment. This reaction moment aims to restore an upright posture. The regulation of the moment is then distributed on three control strategies which are a *ground reaction force control*, *model ZMP control* and *foot landing position control*. A walking controller that is based on the full robot model was developed in [9] for the biped robot JOHNNIE. The authors used a feedback-linearization technique in order to impose a linear behavior for the tracking errors. Buschmann et al. [10] presents a stabilizer that first modifies desired contact forces and torques and second applies a hybrid position/force control which generates local taskspace modifications. Two similar approaches for a whole-body motion controller which also considers long term stability based on the LIPM are presented in [11], [12]. The stabilizer solves a quadratic programming problem for the overall multibody dynamics at each time step which considers e.g. dynamic constraints of the contact forces and joint torques.

One strategy that consists of adaptations of future motion is the sensor based calculation of next foot steps. For the stabilizing step length modifications there are several approaches that apply heuristics or linear models [13], [14], [15]. The capture point introduced by [13] is an often applied method for footstep placement and bipedal walking control

¹Institute of Applied Mechanics, Technische Universität München, 85748 Garching, Germany. E-mail: robert.wittmann@tum.de

[16]. Online model predictive control methods to calculate a CoM trajectory and optimize the next footsteps using the LIPM are presented in [17], [18], [19]. The authors of [17] formulate the optimal control problem for the pendulum in a different way by choosing the time derivative of the CoP as input and setting the weight of the input in the cost function to zero. This allows to give an explicit solution of the problem and to compute more than hundred iterations of the optimization in each control cycle. To our knowledge this is the only work that includes an additional step time optimization.

Even though these approaches present convincing results in experiments we believe that methods using an optimization together with more accurate models are necessary to improve the overall robustness of bipedal walking robots in arbitrary unknown situations. Our model directly resolves the unilateral contacts and has two passive DoFs. This offers new possibilities for motion generation in rough terrain situations or under large disturbances as will be discussed later.

III. SYSTEM OVERVIEW AND PROBLEM DESCRIPTION

This section gives a short overview of the dynamics of bipedal locomotion and describes the bipedal robot LOLA and its walking control system. This information will be used in the next sections to state the main goal of our proposed bipedal walking stabilization.

A. Problem Description

Bipedal robots represent a class of mechanical systems with different challenging properties for planning and control. Beside their nonlinear multi-body dynamics with many degrees of freedom (DoFs) these robots are, in contrast to industrial manipulators, not fixed to the environment. This is a necessary property as bipedal locomotion requires a system that can change periodically the contact states of the feet between open and closed. Splitting the overall DoFs $\mathbf{q}_R \in \mathbb{R}^n$ of the robot into the free floating base located at the torso $\mathbf{q}_T \in \mathbb{R}^6$ and the joints $\mathbf{q}_J \in \mathbb{R}^{n-6}$ the partitioned [20] equations of motion (EoM) of the robot yield

$$\begin{bmatrix} \mathbf{M}_{TT} & \mathbf{M}_{TJ} \\ \mathbf{M}_{JT} & \mathbf{M}_{JJ} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_T \\ \ddot{\mathbf{q}}_J \end{bmatrix} + \begin{bmatrix} \mathbf{h}_T \\ \mathbf{h}_J \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_{\lambda,T}^T \\ \mathbf{J}_{\lambda,J}^T \end{bmatrix} \boldsymbol{\Lambda}. \quad (1)$$

The parts of the mass matrix are denoted by \mathbf{M}_{ij} , \mathbf{h}_i are the nonlinear vectors of Coriolis, centrifugal and gravitational forces and $\boldsymbol{\tau} \in \mathbb{R}^{n-6}$ are the joint torques. The contact forces and moments $\boldsymbol{\Lambda} \in \mathbb{R}^{12}$ are projected with the matrices $\mathbf{J}_{\lambda,T}$, $\mathbf{J}_{\lambda,J}$ to the directions of \mathbf{q}_T and \mathbf{q}_J respectively. With $\boldsymbol{\tau}$ being the system input, the \mathbf{q}_T can not be directly controlled and the robot described by (1) is under-actuated. Physically feasible ground forces are limited because of the unilateral contact, resulting in inequality constraints. They have to be accounted for in order to generate a feasible motion. Examples of such motion generation methods can be found in [21], [22]. But even for perfect tracking (of \mathbf{q}_J) of a feasible motion there are always deviations from the ideal planned state (especially \mathbf{q}_T). Reasons are modeling errors, a finite compliance in the contacts between robot and

the ground and other external disturbances. One strategy to control \mathbf{q}_T is to modify the contact forces $\boldsymbol{\Lambda}$ by changing the joint trajectories \mathbf{q}_J . There were many different methods developed in order to achieve this modification, e.g. changing the robot's CoM motion [7], total angular momentum, foot orientation [23] or horizontal foot positions [4].

The following part gives an overview of the walking control system of LOLA. Referring to the problem description the walking pattern generation and the control strategy are described.

B. System Overview

The experimental platform for this research is the biped robot LOLA shown in Fig. 1. It weights 60 kg, its total height is 1.8m and has 24 electrically actuated joints. The joint configuration is depicted on the right hand side of Fig. 1. The robot is equipped with an Inertial Measurement Unit (IMU) at the upper body and 6-axis force-torque sensors (FTS) located at each foot. The IMU consists of three fiber-optic gyroscopes and three MEMS accelerometers. The system includes internal sensor fusion algorithms which provides accurate and drift-free measurements for the absolute orientation and rotation rate. Those quantities are converted into absolute inclination φ_m and inclination rate $\dot{\varphi}_m$ which are the angles between the upper body and the gravity vector. Details of the mechatronic design can be found in [24].

The control system has a hierarchical design shown in Fig. 2. User commands via joystick or step parameter input are translated into an ideal step sequence which is then used in the trajectory planning. The generated ideal walking pattern in task-space \mathbf{w}_{id} consists of the center of mass (CoM) trajectory, upper body orientation, the pose of both feet and the toe angles. The task-space trajectories are used as set points in the local stabilization. It is based on a hybrid force-position control that performs modifications in task-space depending on sensor data. IMU and FTS measurements are used to modify the commanded stance foot orientation and vertical position. Finally, desired joint angles \mathbf{q}_d and velocities $\dot{\mathbf{q}}_d$ are calculated by inverse kinematics (redundancy resolution is done by *automatic supervisory control* originally formulated by Liégeois [25]). These are then passed on to the local joint controllers. Following a decentralized concept, the local joint feedback loops run on distributed controllers at high sampling rates (50 μ s current, 100 μ s velocity and position), while the central control runs at 1 ms sampling rate on an embedded system. A detailed overview of the walking controller is provided in [26].

In this work we present a method to modify the horizontal CoM motion as well as next foot positions in a model predictive control scheme. The main input is the estimated upper body inclination which is a subset of the unactuated DoFs \mathbf{q}_T . The algorithm has to consider constraints for the CoM motion due to the unilateral contact forces which themselves depend again on the foot positions. The method extends our previously published algorithm [4] for optimizing next footsteps by an optimization of the robot's horizontal CoM trajectories.

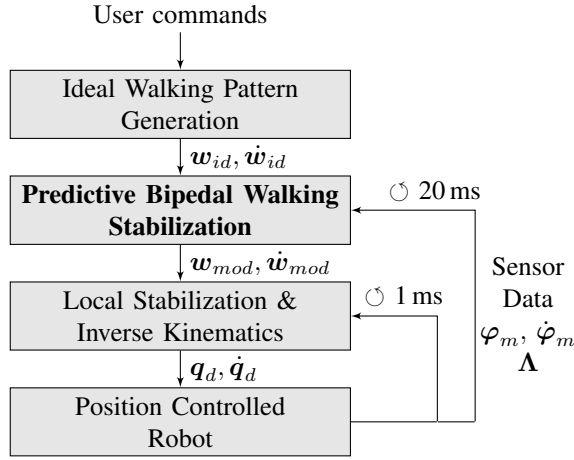


Fig. 2: Overview of the walking control system.

IV. PREDICTIVE BIPEDAL WALKING STABILIZATION

A. Method Overview

The main goal of the presented method is to stabilize the absolute inclination of the robot w.r.t. the ground. It is included in the unactuated DoF of (1). The measurement data of the IMU ($\varphi_m, \dot{\varphi}_m$) is filtered in a state observer [27] in order to estimate the absolute inclination in x- and y-direction ($\hat{\varphi}_x, \hat{\varphi}_y$). Following this approach all disturbances that are not related to joint tracking errors have to be visible in the inclination errors. For rough terrain locomotion, pushes and modeling errors, this turned out to be sufficient information. Other state variables that could be used for feedback are the remaining part of \mathbf{q}_T or the states of all contacts at the feet.

The predictive stabilizer modifies ideal horizontal CoM trajectories, and the overall swing foot trajectories (x, y, z position and orientation about x- and y-axis $\Delta c_x, \Delta c_y$ ¹) producing the modified task-space motion \mathbf{w}_{mod} in Fig. 2. The horizontal CoM motion and final swing foot position are optimized while the vertical swing foot location and $\Delta c_x, \Delta c_y$ are chosen such that early or late contact are avoided and the swing foot touches the ground horizontally (cf. [28], [4]). We choose to use a dynamic model of the biped that includes the mentioned unactuated DoFs of \mathbf{q}_T naturally as well as the CoM and swing foot motion. This enables constructing a method to relate the optimization variables with the control goal. In the following the trajectory modification for the CoM acceleration is denoted by $u_x(t), u_y(t)$ in x- and y-direction and the modification of the next final swing foot location by $\Delta L_x, \Delta L_y$. A planar model is used and the problem is solved in a decoupled way for both directions. Consequently all relations shown below are only described for one direction and can be easily applied for sagittal and lateral motion.

The goal of the presented stabilizer can be formulated

¹refer to Fig. 1 for the global coordinate system definition

mathematically as a minimization of a quadratic cost function

$$J = \Delta \hat{\mathbf{x}}_f^T \mathbf{S}_x \Delta \hat{\mathbf{x}}_f + \int_{t_0}^{t_f} \Delta \hat{\mathbf{x}}^T \mathbf{Q} \Delta \hat{\mathbf{x}} + R u^2 dt \quad (2)$$

for a certain time horizon $t \in [t_0, t_f]$. The state error $\Delta \hat{\mathbf{x}}$ is the difference between ideal and predicted state of the model and will be defined in the next part. Note that the cost function weights \mathbf{S}_x, \mathbf{Q} and R will be parameterized to mainly minimize the absolute inclination error over a certain time horizon which is included in the state variable $\hat{\mathbf{x}}$.

B. Dynamic Prediction Model

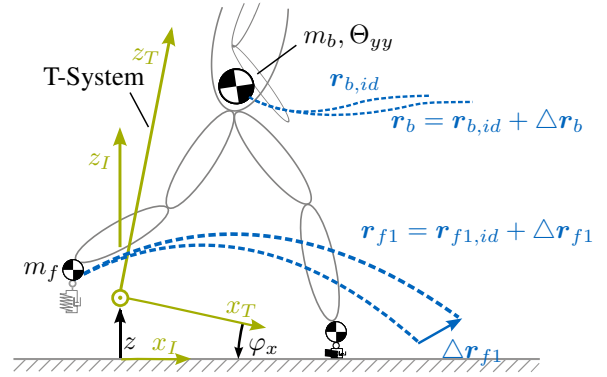


Fig. 3: Prediction model with three point masses, upper body modification Δr_b and swing foot modification Δr_{f1} .

The model proposed in [1] is motivated by the observation that a more accurate model for stiff position controlled bipedal robots has to include the unactuated DoFs between robot and ground. Therefore it has to consider the unilateral and compliant contacts. Starting from the full multi-body model of a bipedal robot (1) the following simplifications are done. The planned CoM and foot trajectories are assumed to be perfectly tracked in the robot's planning frame of reference (FoR) which rotates with the upper body (T-system). Inertia effects are approximately included via three point masses. The unactuated DoFs are the inclination φ_x in x-direction and the vertical translation z (horizontal translation is neglected). They describe the transformation from an inertial (index I, x_I, z_I) to a FoR rotating with the upper body of the robot (index T, x_T, z_T). Furthermore, the trajectories for the robot's upper body $T\mathbf{r}_b$ and the feet $T\mathbf{r}_{f1}, T\mathbf{r}_{f2}$ correspond to the ideal planned trajectories from the walking pattern generation in the T-system. The upper body has the mass m_b and the inertia Θ_{yy} . Each foot has a point mass m_f and one point contact located at the middle of the foot. Those masses are not fixed to each other and follow known trajectories which means that they are time varying but no DoF. The contacts are linear spring-damper pairs (stiffness k_c , damping d_c) with the values identified from the real robot's rubber sole. They act only in z_I -direction and have to be considered as unilateral in order to simulate a walking cycle that necessarily requires switching of the contact states between open and closed. This enables

the model to predict a divergence of φ_x . For a given foot trajectory ${}^T\mathbf{r}_{f_i}(t) = [x_{f_i}(t), 0, z_{f_i}(t)]^T$ and state vector $\mathbf{q} = [z, \varphi_x]^T$, the vertical contact distance is calculated

$$\Delta z_i(\mathbf{q}) = z - x_{f_i}(t) \sin \varphi_x + z_{f_i}(t) \cos \varphi_x \quad (3)$$

During integration (3) has to be compared with a neutral spring length z_0 for each contact in order to set the corresponding contact active or inactive. With the inertial position of upper body and feet obtained by a translation (z) and rotation about the y-axis ($\mathbf{A}_y(-\varphi_x)$)

$${}^T\mathbf{r}_j(\mathbf{q}) = [0, 0, z]^T + \mathbf{A}_y(-\varphi_x) {}^T\mathbf{r}_j, \quad \forall j \in [b, f1, f2] \quad (4)$$

the overall equations of motion of the model in Fig. 3 can be stated in the form

$$\mathbf{M}_p(\mathbf{q}, t)\ddot{\mathbf{q}} + \mathbf{h}_p(\mathbf{q}, \dot{\mathbf{q}}, t) = \boldsymbol{\lambda}_p(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{T}_s \quad (5)$$

where \mathbf{M}_p is the mass matrix of the prediction model, \mathbf{h}_p the nonlinear terms and $\boldsymbol{\lambda}_p$ is the sum of all active contacts N_c projected on the model's DoFs

$$\boldsymbol{\lambda}_p(\mathbf{q}, \dot{\mathbf{q}}, t) = \sum_{i \in N_c} \frac{\partial \Delta z_i}{\partial \mathbf{q}} (-k_{c,i} \Delta z_i - d_{c,i} \Delta \dot{z}_i). \quad (6)$$

$\mathbf{T}_s = [0, \text{sat}(-K_p \Delta \varphi_x - K_d \Delta \dot{\varphi}_x)]^T$ describes the effect of the local stabilization term for the model and includes an approximation of the rotational stiffness for finite sized feet. The gains K_p and K_d are manually tuned for a good approximation. Using the two-DoF-model ensures that a prediction can be done in real-time for a sufficient long time horizon. Predicting 0.8s takes approx. $4 \cdot 10^{-5}$ s using an explicit Euler integrator with a fixed time step of 3 ms.²

The model considers the overall modified motion due to the parameter ΔL_x and the trajectory $u_x(t)$ which is described in the following. The footstep modification is used to calculate a quintic polynomial that begins at current position and ends at ΔL_x . This is shown in Fig. 3 where the polynomial $\Delta {}^T\mathbf{r}_{f1}(\Delta L_x)$ is added to the ideal trajectory ${}^T\mathbf{r}_{f1,id}$ via

$${}^T\mathbf{r}_{f1} = {}^T\mathbf{r}_{f1,id} + \Delta {}^T\mathbf{r}_{f1}(\Delta L_x) \quad (7)$$

which includes a modification of the swing foot height too (for inclination compensation). CoM modifications are added the same way

$${}^T\mathbf{r}_b = {}^T\mathbf{r}_{b,id} + \Delta {}^T\mathbf{r}_b = \begin{bmatrix} {}^T x_{b,id} \\ 0 \\ {}^T z_{b,id} \end{bmatrix} + \begin{bmatrix} \Delta {}^T x_b \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

with the difference that the overall trajectory is modified. We choose to define our input to be the upper body acceleration $u_x = \Delta {}^T \ddot{x}_b$ in order to ensure C^1 -continuity of the generated trajectory. The overall set of first order differential

equations including a double integrator for u_x can be written as

$$\dot{\hat{\mathbf{x}}} = \frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \\ \Delta {}^T x_b \\ \Delta {}^T \dot{x}_b \end{bmatrix} = \begin{bmatrix} \mathbf{M}_p^{-1} [\boldsymbol{\lambda}_p + \mathbf{T}_s - \mathbf{h}_p] \\ \Delta {}^T \dot{x}_b \\ u_x \end{bmatrix} \quad (9)$$

$$= \mathbf{f}(\hat{\mathbf{x}}, t, \Delta L_x, u_x).$$

The state error in (2) will be defined for our implementation as $\Delta \hat{\mathbf{x}} = \hat{\mathbf{x}} - [0, 0, 0, 0, \Delta L_x, 0]^T$. This model does not only improve prediction accuracy when compared to the LIPM, it also has two additional advantages:

- 1) The inclination error which is used as feedback variable is not contained in the commanded trajectories \mathbf{w} which avoids discontinuities in $\Delta x_b(t)$. This is often a problem in generated trajectories together with real-time model predictive control.
- 2) As the contact state is directly resolved and no ZMP-like planning is performed the method can be applied for situations that violate the assumptions of flat foot gaits and horizontal ground.

The next section deals with the real-time minimization of (2) submitted to (9) for $\Delta L_x, u_x$.

C. Real-Time Solution by Conjugate Gradient Algorithm

The resulting dynamic optimization problem will be solved with the conjugate gradient method which was originally proposed by [29]. The author of [30] extended the method for optimal control problems with additional free parameters. The conjugate gradient provides better results than using only the gradient and is at the same time computationally not as expensive as second order algorithms [31] with better stability properties. Additionally it is often used for real-time applications as it can be aborted after each iteration providing a suboptimal but feasible solution (which is not necessarily the fact for e.g. collocation methods [32]). Finding the optimal solution $\mathbf{u}(t) = u_x(t)$ and $\mathbf{p} = \Delta L_x$ for the cost function defined in (2) with the model (9) can be stated as mathematical problem as follows

$$\min_{\mathbf{p}, \mathbf{u}(t)} J = \min_{\mathbf{p}, \mathbf{u}(t)} \left[s(\hat{\mathbf{x}}(t_f), \mathbf{p}) + \int_{t_0}^{t_f} h(\hat{\mathbf{x}}, \mathbf{p}, \mathbf{u}, t) dt \right] \quad (10)$$

$$\text{s.t. } \dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{p}, \mathbf{u}(t), t), \quad \hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0 \quad (11)$$

$$\mathbf{u}(t) \in \mathbf{U}, \quad \mathbf{p} \in \mathbf{P} \quad (12)$$

If the final state $\hat{\mathbf{x}}_f$ is free, the application of Pontryagin's Minimum Principle results in a decoupled boundary value problem (see Appendix, (24) and (25)). It is solved with the conjugate gradient method iteratively by integrating the state $\hat{\mathbf{x}}$ from t_0 to t_f forward and the costate $\boldsymbol{\psi}$ backward in time. Afterward an update of the control input $\mathbf{u}(t)$ and parameter \mathbf{p} for the (k+1)-th iteration is generated

$$\mathbf{u}^{k+1}(t) = \mathbf{u}^k(t) + \alpha^k \mathbf{s}_u^k(t) \quad (13)$$

$$\mathbf{p}^{k+1} = \mathbf{p}^k + \alpha^k \mathbf{s}_p^k. \quad (14)$$

²The onboard computer runs the RT-OS QNX 6.6 and is equipped with an Intel core i7-4770@3.4 Ghz and 8Gb Ram

The conjugate gradient directions $\mathbf{s}_u^k(t)$ and \mathbf{s}_p^k are determined by the Fletcher-Reeves adaptation

$$\mathbf{s}_u^k(t) = -\mathbf{g}_u^k(t) + \beta_u^k \mathbf{s}_u^{k-1}(t), \quad \beta_u^k = \frac{(\mathbf{g}_u^k, \mathbf{g}_u^k)}{(\mathbf{g}_u^{k-1}, \mathbf{g}_u^{k-1})} \quad (15)$$

$$\mathbf{s}_p^k = -\mathbf{g}_p^k + \beta_p^k \mathbf{s}_p^{k-1}, \quad \beta_p^k = \frac{(\mathbf{g}_p^k)^T \mathbf{g}_p^k}{(\mathbf{g}_p^{k-1})^T \mathbf{g}_p^{k-1}} \quad (16)$$

with $(\mathbf{g}_u^k, \mathbf{g}_u^k) = \int_{t_0}^{t_f} \mathbf{g}_u^k(t), \mathbf{g}_u^k(t) dt$ using the gradients

$$\mathbf{g}_u^k(t) = \frac{\partial H}{\partial \mathbf{u}} \quad (17)$$

and (from Appendix relation (27))

$$\mathbf{g}_p^k = \left. \frac{\partial s}{\partial \mathbf{p}} \right|_{t_0} + \int_{t_0}^{t_f} \frac{\partial H}{\partial \mathbf{p}} dt \quad (18)$$

for the unconstrained case. They have to be replaced by a projection pointing into the normal cone of \mathcal{U}, \mathcal{P} if the boundary of the valid regions is reached. The factor α^k is computed from the one-dimensional line search problem

$$\alpha^k \leftarrow \min_{\alpha} J(\mathbf{u}^k + \alpha \mathbf{s}_u^k, \mathbf{p}^k + \alpha \mathbf{s}_p^k) \quad (19)$$

which we solve approximately using the backtracking method. The overall procedure is summarized in Algorithm 1.

Algorithm 1 Conjugate Gradient Method

- 1: Determine $\hat{\mathbf{x}}_a$
 - 2: initialize $u_x^0(t) \leftarrow u_{x,previous}^*(t)$
 - 3: initialize ΔL_x^0
 - 4: $k \leftarrow 0$
 - 5: **repeat**
 - 6: solve $\hat{\mathbf{x}}^k(\mathbf{u}^k, \mathbf{p}^k, t)$ from (11)
 - 7: solve $\psi^k(\mathbf{u}^k, \mathbf{p}^k, \hat{\mathbf{x}}^k, t)$ from (25)
 - 8: compute $\mathbf{s}_u^k, \mathbf{s}_p^k$ from (15) and (16)
 - 9: solve line search (19)
 - 10: update $\mathbf{u}^{k+1}, \mathbf{p}^{k+1}$ with (13) and (14)
 - 11: $k \leftarrow k + 1$
 - 12: **until** (converged or $k > \text{max iterations}$)
 - 13: **end**
-

D. System Integration Details

The algorithm requires an initial value $\hat{\mathbf{x}}_a$ and an initial solution for $\mathbf{u}^0(t), \mathbf{p}^0$. The former is obtained by feeding the IMU-measurements $(\varphi_m, \dot{\varphi}_m)$ into a state observer. The observer is similar to the one presented in [27] but uses the nonlinear model (5). It is based on an extended kalman filter with a compensation for model errors and external disturbances. The overall predictive bipedal walking stabilization is called every 20 ms in the walking control system. For all experiments the optimization horizon $T_{opt} = t_e - t_a$ is set to the time of one step which we choose as 0.8 s for this paper. The initial solution for the control trajectory u_x^0 uses the

solution from previous optimization $u_{x,previous}$ following a receding horizon approach

$$u_x^0(\tau) = u_{x,previous}(\tau - 20 \text{ ms}) \quad (20)$$

The value ΔL_x^0 is computed from the predicted CoM position error $\Delta x_c(t_1)$ and velocity error $\Delta \dot{x}_c(t_1)$ at the end of current step (t_1) by the heuristic motivated by the divergent component of motion of the LIPM

$$\Delta L_x^0 = k_1 \left(\Delta x_c(t_1) + \sqrt{z_c/g} \Delta \dot{x}_c(t_1) \right). \quad (21)$$

The overall planned CoM height is described by z_c and $k_1 \in [0, 1]$ is a manually tuned factor.

At the same time the predicted state in x- and y-direction at time t_1 is used in order to compute modification values for the final foot rotation $(\Delta c_x, \Delta c_y)$ and height (Δz) as described in [4]. This helps to compensate for expected inclination errors.

All mentioned variables can be summarized in the vector $\mathbf{\Gamma} = [\Delta L_x, \Delta L_y, \Delta c_x, \Delta c_y, \Delta z]$ which is used to calculate a set of fifth-order polynomials for the task-space modifications of the legs

$$\Delta \mathbf{w}_{legs}(t) = \mathbf{f}(t, \mathbf{\Gamma}) \quad (22)$$

continuously when new values $\mathbf{\Gamma}$ are generated. The CoM modification $\Delta \mathbf{w}_{com}(t)$ is computed by integrating the accelerations $u_x(t), u_y(t)$ in both directions twice. Using only the computed accelerations avoids the problem of generating discontinuous trajectories $\Delta \mathbf{w}_{com}$ and $\Delta \dot{\mathbf{w}}_{com}$. A similar strategy for online generated task-space velocities is used in [33]. The final modified task-space position for a time instance t_k is computed via

$$\mathbf{w}_{mod}(t_k) = \mathbf{w}_{id}(t_k) + \begin{bmatrix} \Delta \mathbf{w}_{legs}(t_k) \\ \Delta \mathbf{w}_{com}(t_k) \end{bmatrix} \quad (23)$$

and analogous for the task-space velocity $\dot{\mathbf{w}}_{mod}(t_k)$. The overall computational time for the whole predictive stabilization algorithm is below 3.5 ms for one direction. The maximum number of iterations is set to three for all experiments shown below.

V. RESULTS

A. Simulation with Disturbances

The proposed method was implemented in the control system of LOLA and tested in our simulation environment. It considers compliant contacts as well as the motor dynamics and the joint control loops. The robot is stepping in place while it is pushed at the upper body with a force acting in x- and y-direction. The unknown disturbance force is shown in Fig. 4. The plot also shows the resulting inclination errors of the simulated robot and compares it with the result when the predictive stabilizer is turned off. It can be seen that the absolute inclination of the robot does not diverge with the predictive stabilizer.

The total CoM modification trajectory and the result from single optimizations are depicted in Fig. 5. At the beginning of the disturbance the method requires several calls in

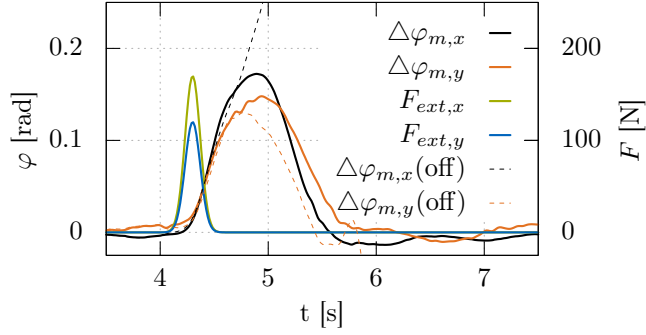


Fig. 4: Disturbance forces and inclination errors of the simulated robot with the proposed stabilization method and for the method turned off.

order to adapt to the unknown situation. The corresponding trajectories for the x- and y-positions of the feet can be seen in Fig. 6. The effect of choosing different prediction horizons T_{opt} is underlined in Fig. 7. The same simulation as described above is performed with three different horizons (0.72s, 0.8s, 0.96s) and it can be seen that the stabilizer performs better with higher prediction horizons.

VI. CONCLUSION AND OUTLOOK

A new predictive method to determine a stabilizing motion adaptation of the robot is presented in this work. It uses the upper body inclination and inclination rate in order to optimize CoM and foot trajectories. A previously proposed nonlinear model of the biped robot is used in a model predictive optimization. The resulting problem is solved with a conjugate gradient algorithm that ensures computational times that allow an application in real-time. Results from our simulation environment are presented where the method is integrated into the walking control system of the bipedal robot LOLA. Next steps include finishing testing the method on the real robot which performs very close to the simulated robot. The planned experiments are walking while the robot receives external pushes and walking over uneven terrain.

APPENDIX

The conditions from Pontryagin's Minimum Principle for the problem (10) to (12) can be derived by stating the augmented cost function

$$\bar{J} = s(\mathbf{y}(t_f), \mathbf{p}) + \int_{t_0}^{t_f} h(\mathbf{y}, \mathbf{p}, \mathbf{u}, t) + \psi^T(\mathbf{f}(\mathbf{y}, \mathbf{p}, \mathbf{u}, t) - \dot{\mathbf{y}}) dt + \psi_0^T(\mathbf{y}_0 - \mathbf{y}(t_0))$$

with the Lagrange multipliers $\psi(t)$ and ψ_0 . Using the Hamiltonian $H(\mathbf{y}, \mathbf{p}, \mathbf{u}, t, \psi) = h(\mathbf{y}, \mathbf{p}, \mathbf{u}, t) + \psi^T \mathbf{f}(\mathbf{y}, \mathbf{p}, \mathbf{u}, t)$ the first variation is

$$\delta \bar{J} = \frac{\partial s}{\partial \mathbf{y}} \Big|_{t_f} \delta \mathbf{y}(t_f) + \frac{\partial s}{\partial \mathbf{p}} \Big|_{t_f} \delta \mathbf{p} + \delta \psi_0^T (\mathbf{y}_0 - \mathbf{y}(t_0)) + \psi_0 \cdot \mathbf{0} + \int_{t_0}^{t_f} \frac{\partial H}{\partial \mathbf{y}} \delta \mathbf{y} + \frac{\partial H}{\partial \mathbf{p}} \delta \mathbf{p} + \frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u} + \frac{\partial H}{\partial \psi} \delta \psi - \delta \psi^T \dot{\mathbf{y}} - \psi^T \delta \dot{\mathbf{y}} dt.$$

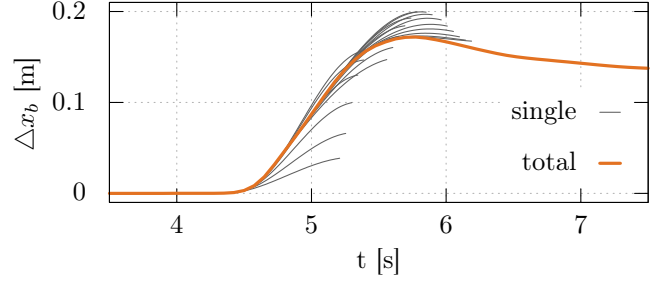


Fig. 5: CoM modification trajectories in simulation for single optimizations (single) and the overall executed trajectory (total).

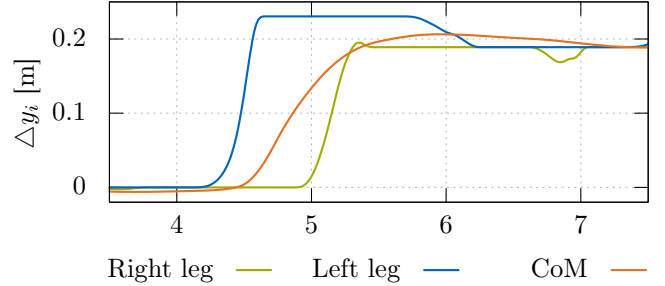
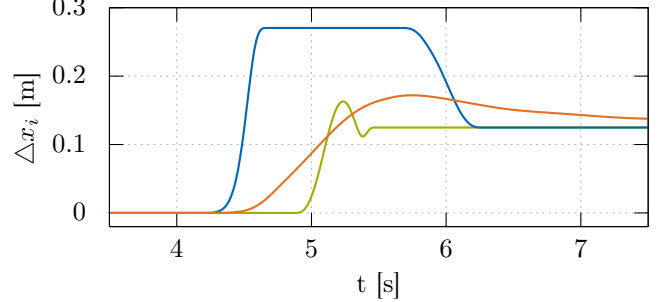


Fig. 6: Modification trajectories for the legs and the CoM in x- and y-direction.

Integrating by parts $\int \psi^T \delta \dot{\mathbf{y}}$ and rearranging leads to

$$\begin{aligned} \delta \bar{J} &= \left(\frac{\partial s}{\partial \mathbf{y}} \Big|_{t_f} - \psi(t_f)^T \right) \delta \mathbf{y}(t_f) + \delta \psi_0^T (\mathbf{y}_0 - \mathbf{y}(t_0)) \\ &+ \int_{t_0}^{t_f} \left(\frac{\partial H}{\partial \mathbf{y}} + \dot{\psi}^T \right) \delta \mathbf{y} + \left(\frac{\partial H}{\partial \psi} - \dot{\mathbf{y}}^T \right) \delta \psi + \frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u} dt \\ &+ \frac{\partial s}{\partial \mathbf{p}} \Big|_{t_f} \delta \mathbf{p} + \int_{t_0}^{t_f} \frac{\partial H}{\partial \mathbf{p}} \delta \mathbf{p} dt \end{aligned}$$

The conditions for the optimal solution are finally

$$\frac{\partial s}{\partial \mathbf{y}} \Big|_{t_f} - \psi(t_f)^T = 0 \quad \mathbf{y}_0 - \mathbf{y}(t_0) = 0 \quad (24)$$

$$\frac{\partial H}{\partial \mathbf{y}} + \dot{\psi}^T = 0 \quad \frac{\partial H}{\partial \psi} - \dot{\mathbf{y}}^T = 0 \quad (25)$$

$$\frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u} = 0 \quad (26)$$

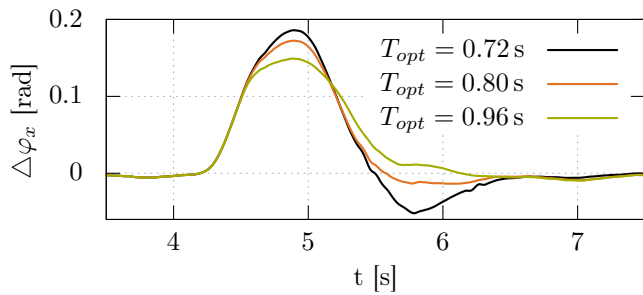


Fig. 7: Resulting upper body inclination error of the biped for three different time horizons in the optimization.

$$\left(\frac{\partial s}{\partial \mathbf{p}} \Big|_{t_f} + \int_{t_0}^{t_f} \frac{\partial H}{\partial \mathbf{p}} dt \right) \delta \mathbf{p} = 0. \quad (27)$$

These are the same conditions as for an optimal control problem with free final state and fixed final time with the additional condition (27) for the free parameters \mathbf{p} .

ACKNOWLEDGMENT

This work is supported by the Deutsche Forschungsgemeinschaft (project BU 2736/1-1).

REFERENCES

- [1] R. Wittmann, A.-C. Hildebrandt, A. Ewald, and T. Buschmann, "An Estimation Model for Footstep Modifications of Biped Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2014, pp. 2572–2578.
- [2] A.-C. Hildebrandt, D. Wahrmann, R. Wittmann, D. Rixen, and T. Buschmann, "Real-Time Pattern Generation Among Obstacles for Biped Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2015, pp. 2780–2786.
- [3] D. Wahrmann, A.-C. Hildebrandt, R. Wittmann, D. Rixen, and T. Buschmann, "Fast Object Approximation for Real-Time 3D Obstacle Avoidance with Biped Robots," *IEEE International Conference on Advanced Intelligent Mechatronics*, 2016.
- [4] R. Wittmann, A.-C. Hildebrandt, D. Wahrmann, D. Rixen, and T. Buschmann, "Real-Time Nonlinear Model Predictive Footstep Optimization for Biped Robots," in *IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 711–717.
- [5] K. Nishiwaki and S. Kagami, "Online Walking Control System for Humanoids with Short Cycle Pattern Generation," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 729–742, may 2009.
- [6] —, "Frequent walking pattern generation that uses estimated actual posture for robust walking control," in *IEEE/RAS International Conference on Humanoid Robots*. IEEE, dec 2009, pp. 535–541.
- [7] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of Honda humanoid robot," in *IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1998, pp. 1321–1326.
- [8] T. Takenaka, T. Matsumoto, T. Yoshiike, T. Hasegawa, S. Shirokura, H. Kaneko, and A. Orita, "Real time motion generation and control for biped robot -4th report: Integrated balance control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct 2009, pp. 1601–1608.
- [9] K. Löffler, M. Gienger, and F. Pfeiffer, "Model based control of a biped robot," in *7th International Workshop on Advanced Motion Control*. IEEE, 2002, pp. 443–448.
- [10] T. Buschmann, "Simulation and Control of Biped Walking Robots," Ph.D. dissertation, 2010.
- [11] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *IEEE International Conference on Robotics and Automation*. IEEE, may 2014, pp. 2589–2594.
- [12] A. Sherikov, D. Dimitrov, and P.-b. Wieber, "Whole body motion controller with long-term balance constraints," in *IEEE/RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 444–450.

- [13] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *IEEE/RAS International Conference on Humanoid Robots*. IEEE, dec 2006, pp. 200–207.
- [14] J. Rebuta, F. Canas, J. Pratt, and A. Goswami, "Learning Capture Points for humanoid push recovery," in *IEEE/RAS International Conference on Humanoid Robots*. IEEE, nov 2007, pp. 65–72.
- [15] J. Hodgins and M. Raibert, "Adjusting step length for rough terrain locomotion," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 289–298, jun 1991.
- [16] J. Engelsberger and C. Ott, "Integration of vertical COM motion and angular momentum in an extended Capture Point tracking controller for bipedal walking," in *IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 183–189.
- [17] J. Urata, K. Nishiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba, "Online decision of foot placement using singular LQ preview regulation," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, oct 2011, pp. 13–18.
- [18] R. Tajima, D. Honda, and K. Suga, "Fast running experiments involving a humanoid robot," in *IEEE International Conference on Robotics and Automation*. IEEE, may 2009, pp. 1571–1576.
- [19] P.-B. Wieber, "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, dec 2006, pp. 137–142.
- [20] Y. Fujimoto, S. Obata, and A. Kawamura, "Robust biped walking with active interaction control between foot and ground," in *IEEE International Conference on Robotics and Automation*, vol. 3, no. May. IEEE, 1998, pp. 2030–2035.
- [21] T. Buschmann, S. Lohmeier, M. Bachmayer, H. Ulbrich, and F. Pfeiffer, "A collocation method for real-time walking pattern generation," in *IEEE/RAS International Conference on Humanoid Robots*. IEEE, nov 2007, pp. 1–6.
- [22] S. Kajita, O. Matsumoto, and M. Saigo, "Real-time 3D walking pattern generation for a biped robot with telescopic legs," in *IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 2001, pp. 2299–2306.
- [23] T. Buschmann, S. Lohmeier, and H. Ulbrich, "Biped walking control based on hybrid position/force control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct 2009, pp. 3019–3024.
- [24] S. Lohmeier, "System design and control of anthropomorphic walking robot LOLA," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 6, pp. 658–666, 2009.
- [25] A. Liégeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.
- [26] T. Buschmann, V. Favot, S. Lohmeier, M. Schwienbacher, and H. Ulbrich, "Experiments in fast biped walking," in *IEEE International Conference on Mechatronics*. IEEE, apr 2011, pp. 863–868.
- [27] R. Wittmann, A.-C. Hildebrandt, D. Wahrmann, T. Buschmann, and D. Rixen, "State Estimation for Biped Robots Using Multibody Dynamics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2015, pp. 2166–2172.
- [28] K. Nishiwaki and S. Kagami, "Walking control on uneven terrain with short cycle pattern generation," in *IEEE/RAS International Conference on Humanoid Robots*. IEEE, nov 2007, pp. 447–453.
- [29] L. Lasdon, S. Mitter, and A. Waren, "The conjugate gradient method for optimal control problems," *IEEE Transactions on Automatic Control*, vol. 12, no. 2, pp. 132–138, apr 1967.
- [30] M. Boček, "Conjugate gradient algorithm for optimal control problems with parameters," *Kybernetika*, vol. 16, no. 5, pp. 454–461, 1980.
- [31] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4906–4913, oct 2012.
- [32] J. Nocedal and S. J. Wright, "Numerical Optimization," pp. 1–651, 2004.
- [33] K. Nishiwaki, J. Chestnutt, and S. Kagami, "Autonomous Navigation of a Humanoid Robot over Unknown Rough Terrain using a Laser Range Sensor," *The International Journal of Robotics Research*, 2012.