2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)
November 3 - 5, 2015, Seoul, Korea

Real–Time Nonlinear Model Predictive Footstep Optimization for Biped Robots

Robert Wittmann, Arne-Christoph Hildebrandt, Daniel Wahrmann, Daniel Rixen and Thomas Buschmann[1]

***Abstract*— A well known strategy in biped locomotion to prevent falling in the presence of large disturbances is to modify next footstep positions of the robot. Solving this complex control problem for the overall model of the robot is a challenging task. Published methods employ either linear models or heuristics to determine those positions. This paper introduces a new optimization method using a nonlinear and more accurate model of the robot. The resulting optimization problem to calculate the necessary footstep modification is solved by a direct shooting method. Using a problem formulation in an unconstrained way enables an optimization that performs in real–time rates. Further we present our overall framework that uses sensor feedback in trajectory generation. Experimental results of our biped robot LOLA show the effectiveness of the method under real world conditions.**

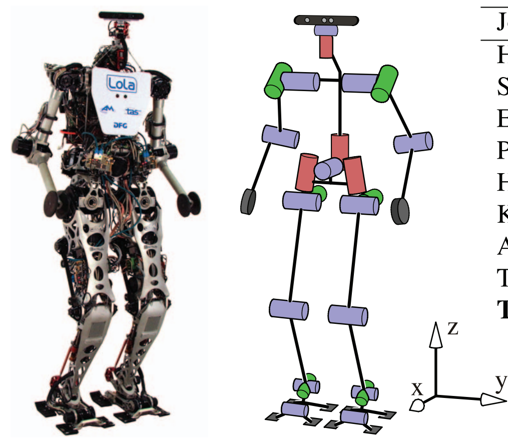
I. INTRODUCTION

Biped robots present a class of mechanical systems with many different challenging properties for planning and control that have to be solved to enable walking such as unilateral contacts, under-actuation, nonlinear multibody dynamics and a large number of degrees of freedom (DoF). In order to bring these robots from controlled laboratory conditions to the real world, the robustness against unknown disturbances has to be increased. A deviation of the robot's current state from the desired behavior has to be detected and the motion generation has to include this state to modify the planned trajectories. The presented approach adds sensor feedback to trajectory generation in a model predictive way which introduces three challenges:

(1) Processing the sensor data with a filter in order to estimate the current state of the robot. The important information concerning global stability has to be extracted.
(2) Prediction the behavior of the robot in the future with an appropriate model. This model has to be sufficiently accurate and fast for a certain time horizon to be applicable in real–time implementations.
(3) Modification the foot trajectory parameters by evaluating the predicted state evolution with the estimated current state. The future step lengths in x– and y–direction are optimized and used to determine the height and angle of the swing foot at the end of current step. This closes the sensor feedback loop to trajectory generation.

This work focuses on the third aspect by using our previously proposed prediction model [1] in an optimization method. The most important parameters are the step lengths for which we formulate a model predictive optimization solved by a direct shooting method.

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



Joint	DoF
Head	2
Shoulder	2
Elbow	1
Pelvis	2
Hip	3
Knee	1
Ankle	2
Toe	1
Total	**24**

Fig. 1. Photograph and kinematic structure of LOLA.

II. RELATED WORK

In contrast to the above mentioned strategy conventional walking controllers perform local modifications of the generated motion. Some examples are discussed in the following paragraph. Buschmann et al. [2] present a stabilization unit that first modifies desired contact forces and torques and second applies hybrid position/force control which generates local task–space modifications. This is realized by an explicit contact model between foot and ground. There are several other works that are based on a stabilization which considers the interaction dynamics between robot and ground [3], [4]. Takenaka et al. [5] accelerate the center of mass in order to stabilize the robot. The presented "model ZMP control" enables the robot to walk on uneven terrain with unmodelled 30 mm high irregularities. A similar approach is presented in [6], [7] where an "auxiliary ZMP" is added to the reference in order to include deviations from the ideal planned trajectories. This matches in their framework that uses "preview control" to calculate a center of mass trajectory. Two similar works for a whole–body motion controller which considers long term stability based on the simple linear inverted pendulum model (LIPM) are presented in [8] and [9]. The stabilizer solves a quadratic programming problem for the overall multibody dynamics at each time step which considers dynamic constraints of the contact forces, joint torques, etc.. Simulation results of the methods are presented.
General frameworks to include sensor feedback to trajectory generation are proposed by Nishiwaki et al. [10], [11] and Tajima et al. [12]. In contrast to above mentioned stabilizers

these methods use the current measured state as initial value for trajectory generation and stabilize it by solving a planning problem with high frequency. Experimental results show biped robots walking on unmodelled rough terrain and in the presence of pushes. The algorithms of how to adjust the step lengths are not described in detail or are based on a linear model. For the calculation of stabilizing step length modifications there are several approaches that apply heuristics or linear models [13], [14], [15], [1]. The authors of [13] present an energy-based method to obtain a stable cyclic motion for a hopping robot. They introduce the so-called neutral point, a place where the system has to step to in order to maintain the current velocity. Stepping before or behind this point accelerates or decelerates the system. [14] combines a linear cart table model with online learning. The initialization is done by an analytic controller. Another method to determine stable footsteps is the so-called Capture Point [15] which is a LIPM-based method and utilizes conservation of the orbital energy. The idea is that the stepping location is chosen such that the CoM will come to rest over this point. There are several other works [16], [17] based on and extending this method. The work in [1] combines a nonlinear model of the robot that is used to predict the state evolution. The resulting prediction is then used in a heuristically update law for the next foot step position. Linear online model predictive control methods to calculate a CoM trajectory and optimize the next footsteps using the LIPM are presented in [18], [19]. Even though these works 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 biped walking robots. In this work we use our previously proposed prediction model to formulate an optimization for the foot step positions. This was especially necessary to apply the method while walking forward with approx. 0.5 m/s during external disturbances. Foot placement is formulated as a model predictive control problem.

In Section III we give a brief overview of the experimental platform LOLA. Section IV reviews the main properties of the prediction model we are employing in this work and which is applied to the direct shooting method described in Section V. Implementation details of the optimization and an optimization example are discussed as well. Section VI shows the integration of the optimization into a real–time sensor feedback framework to our overall control system. The experimental results are shown in Section VII and a conclusion is given in Section VIII.

III. THE BIPED ROBOT LOLA

The experimental platform for this research is the biped robot LOLA shown in Fig. 1. It weights 60 kg, is 1.8 m tall and has 24 electrical actuated joints. The joint configuration is depicted on the right hand side of Fig. 1. Following a decentralized concept, the low-level joint feedback loops run on distributed controllers at high sampling rates (70 µs current, 140 µs velocity), while high-level control runs at 2 ms sampling rate on a central PC. Details to the mechatronic design can be found in [20].

The high–level control has a hierarchical design. Fig. 2 gives a brief overview. A user command via joystick or parameter input is translated into an ideal step sequence which is then used in the ideal trajectory planning. The generated 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 stabilization unit. It is based on a hybrid force–position control that performs local modifications in task–space depending on sensor data. 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 Liégois). These are then processed to the local joint controllers. A detailed overview of the walking controller is provided in [21].

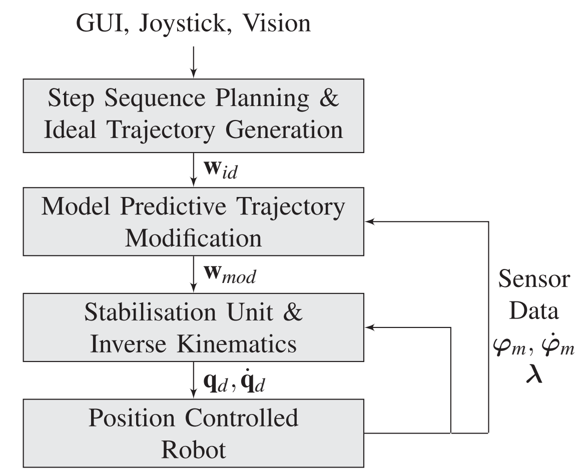


Fig. 2. Overview of the walking control system.

IV. PREDICTION MODEL

This section gives a brief review of our previously proposed prediction model [1]. The main motivation is based on the observation that the (linear) inverted pendulum model does not reliably predict the robot's behavior under large disturbances (also with model extensions like flywheel, torque controller, etc.). We observed that the model has to include the unactuated degrees of freedom (DoF) between robot and ground and that the unilateral and compliant contacts need to be properly accounted for. The planar model in x–direction[1] is shown in Fig. 3. Its passive DoFs are the absolute inclination φ_x of the robot about the y–axis and the absolute translation z. Further the planned (and known) trajectories $\mathbf{r}_{f1,2}$ and $\mathbf{r}_b$ are assumed to be tracked perfectly in the robot's fixed frame of reference (FoR). During the prediction the feet and the robot's upper body follow those trajectories. Each foot is modeled with a point mass m_f and the upper body with the point mass m_b and inertia Θ_{zz}. The contact model of the foot is a point contact which consists of

[1] All further considerations are done in this direction, but are also applied the same way in y–direction by only changing model parameters and the corresponding ideal trajectories.

a linear spring (k_c) and damper (d_c) acting always vertically to the ground. An additional torque T_{stab} is acting on the stance foot in order to integrate a PD-controller for the robot's stabilization. Applying Lagrange's formalism one can

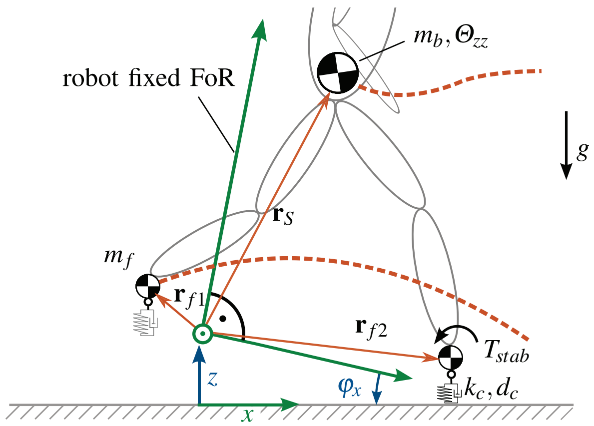


Fig. 3. Prediction model with 2 unactuated DoFs and unilateral compliant contacts (exemplary shown for x–direction).

derive the equations of motion for the system. Introducing $\mathbf{q} = [z, \varphi_x]^T$ they can be written as

$$\mathbf{M}(\mathbf{q},t)\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q},\dot{\mathbf{q}},t) = \boldsymbol{\lambda}(\mathbf{q},\dot{\mathbf{q}},t) \tag{1}$$

with the mass matrix $\mathbf{M}$, vector of Coriolis, centrifugal and gravitational forces $\mathbf{h}$ and generalized (contact) forces $\boldsymbol{\lambda}$. Note that the time dependent variables $\mathbf{r}_{f1,2}(t)$ and $\mathbf{r}_b(t)$ and its derivatives are included in these quantities. The stabilization is realized with the PD–control law

$$T_{stab} = sat(-K_p\varphi_x - K_d\dot{\varphi}_x) \tag{2}$$

where the control gains K_p and K_d are parametrized the same way as on the real robot and *sat* represents a saturation of the output with the geometrical limits of the real foot. Finally the model (1),(2) is transformed to a first order differential equation system

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}},t) \quad \text{with } \hat{\mathbf{x}} = [z, \varphi_x, \dot{z}, \dot{\varphi}_x]^T. \tag{3}$$

The integration of (3) is performed numerically with an explicit Euler–integrator (step size $\triangle t = 2\,\text{ms}$) and the state of the contacts, whether they are closed or open, has to be resolved in each integration step. Integrating the model that way takes about 150 μs for a time horizon of 1.6 s [2].

In contrast to our initial implementation [1], we decided to simplify the foot model (original model: foot length $2l$, two contacts and an additional DoF α which is set by a force controller) in order to speed up the computational time for the prediction. This is necessary as we are using the model in an optimization with several iterations and we have to satisfy real–time requirements. In future work we are planning to increase the run–time performance such that the original proposed prediction model can used.

Using the model it is possible to predict the behaviour of the robot for a measured initial inclination $\varphi_{x,o}$ and translation z_o for a certain time-horizon (1-2 s) considering also the ideal planned trajectories and the effect of the stabilisation unit of the real robot. Additionally the model enables to evaluate trajectory modifications like the foot trajectory. This is essential for the next section where the nonlinear prediction model is used to optimize the swing foot trajectory.

[2]All calculations are done with an Intel i3-2100@3.1Ghz and 8Gb Ram

V. Real–time Nonlinear Model Predictive Footstep Optimization

Using nonlinear models to optimize trajectories in real–time is a difficult task due to run–time restrictions and there is a variety of methods to tackle such problems. Real–time (or online) nonlinear model predictive control with shooting methods has its origin in the field of chemical processes [22]. There is a huge field of research trying to increase speed for those methods in online applications (see e.g. [23] for an overview). Using a variant of classic *Differential Dynamic Programming*, Tassa et al. [24] present a trajectory optimizer performing almost in real–time for the application to a humanoid robot getting up. In contrast to the above mentioned methods, we do not have to optimize a control sequence but instead parametrize the foot trajectory with the desired final position and use a direct shooting method to optimize this single parameter.

A. Problem statement

There are two aspects that have to be considered if one tries to modify the step length according to current state. First the next stepping position can not be changed immediately because the robot needs a finite amount of time to move the foot to the desired position. Second the effect of the modified step length does not act until the current step is finished. The consequences are visualized in Fig. 4, where the robot's absolute inclination is the same for the ideal and optimized trajectories at time t_1 (if the feet's inertia is neglected). Other stabilization strategies can be included in this state prediction. From the end of the first step (t_1) until the second step (t_2) the modified step length has a direct influence to the behavior of the robot and can stabilize it. Using a predictive method one has to include this time interval. These considerations form the basis of our optimization algorithm to determine the optimal step length.

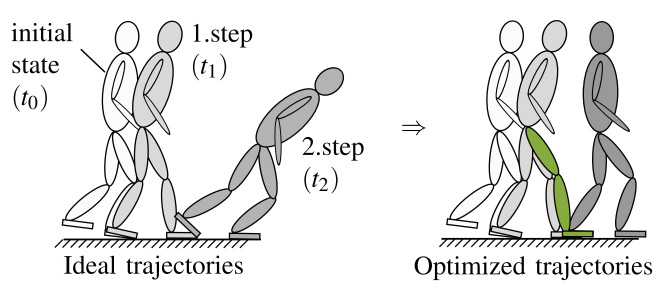


Fig. 4. State evolution of a biped with disturbed initial state (current state) with snapshots at $t = t_0, t_1, t_2$. A possible optimal solution is shown on the right hand side.

The input of the optimization is given by the measured respectively estimated initial state (absolute inclination $\hat{\varphi}_{x,0}$

and inclination rate $\dot{\varphi}_{x,0}$) of the robot. These values are used to initialize the prediction model. The optimization variable is the footstep modification in x–direction $\triangle L_x$ (and in y–direction $\triangle L_y$) for current step. Due to the problem description from Fig. 4 the corresponding time interval is chosen from t_1 to t_2. The problem is formulated as a minimization of the quadratic cost function

$$J_c(\triangle L_x) = \beta_0 \triangle L_x^2 + \int_{t_1}^{t_2} \hat{\mathbf{x}}^T \mathbf{Q} \hat{\mathbf{x}} \,\mathrm{d}t. \tag{4}$$

where $\mathbf{Q}$ is a diagonal matrix to weight the different states. The additional quadratic term of the step modification weighted with β_0 has a regulating effect and helps avoiding a curvature near zero of the cost function. Note that the state evolution for $\hat{\mathbf{x}}(t)$ is determined by integrating (3) which can be extended to depend also on the modified step length

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, t, \triangle L_x). \tag{5}$$

To realize this the footstep modification $\triangle L_x$ is used as final value to calculate a smooth trajectory $\triangle \mathbf{r}_{fi}(\triangle L_x)$ from current set point using a quintic polynomial. The overall trajectory is determined by adding the modification to the ideal planned swing foot trajectory ${}_T\mathbf{r}_{fi} = {}_T\,\mathbf{r}_{fi,id} + \triangle \mathbf{r}_{fi}(\triangle L_x)$. End conditions like $\varphi = 0$ or $\dot{\varphi} = 0$ are omitted in the formulation due to the fact that we want to keep the problem small in order to satisfy real–time computation time and it is not mandatory or possible to come back to an undisturbed state after only one step. If this is the case and the robot can not come back to an undisturbed state after only one step, next footsteps will be modified until the robot reaches the ideal state.

We solve the optimization problem for the parameter $\triangle L_x$ using direct shooting (see [25] for an overview). For a given step length, one can integrate the prediction model from t_1 to t_2 and evaluate the cost function for this horizon. Afterwards an update law is computed with Newton's method

$$\triangle L_x^{k+1} = \triangle L_x^k + \alpha^k \underbrace{\left(\nabla^2_{L_x} J_c\right)^{-1} \nabla_{L_x} J_c}_{:= p^k} \tag{6}$$

using first and second derivatives of the cost function J_c and a scaling factor $\alpha^k \in [0, 1]$ for the search direction p^k. Applying the new step length to the prediction model, it can be integrated again to evaluate the cost function for the updated trajectory.

The remaining part is how to determine the initial state $\hat{\mathbf{x}}(t_1)$ and an initial solution for the step length $\triangle L_x^0$. To this end the prediction model is integrated from t_0 to t_1 (step length modifications have no influence in this time period) which directly delivers the state $\hat{\mathbf{x}}(t_1)$. The initial step length is computed using the predicted error of the CoM position $\triangle x_c(t_1)$ and velocity $\triangle \dot{x}_c(t_1)$ in a linear heuristic

$$\triangle L_x^0 = \alpha_0 \left(\triangle x_c(t_1) + \sqrt{\frac{g}{m}} \triangle \dot{x}_c(t_1) \right) \tag{7}$$

with a manually tuned factor $\alpha_0 \in [0, 1]$, the total robot mass m and gravity acceleration g. This heuristic is motivated by the unstable or divergent solution of the LIPM [26] which is additionally scaled with α_0. The next section gives additional details to the proposed method.

B. Implementation details

The update law (6) requires computation of first and second derivatives of the cost function. Having the problem formulated in Lagrange form (cost function involves an integral), we are using numerical derivatives with the central difference formula to compute these quantities. This requires three integrations of (5) for each iteration of the shooting method. In order to reduce the computation time, the required ideal trajectories are evaluated at the required time steps for the prediction horizon in advance and are stored into an array which is then used in the integration of the prediction model. The weights of the cost function for all experiments are chosen to

$$\beta_0 = 2.0, \quad \mathbf{Q} = \mathrm{diag}(0.01, 3.0, 0.001, 0.01). \tag{8}$$

The value α is determined by using the backtracking approach to solve the line-search problem [27]. Starting with a value $\alpha = 1$ it is decreased with a factor ρ until the condition

$$J_c(\triangle L_x^k + \alpha p^k) \leq J_c(\triangle L_x^k) + c\alpha \left(\nabla_{L_x} J_c\right)^T p^k \tag{9}$$

is satisfied. The constant c determines the necessary reduction of the cost function and is set in our implementation to $c = 0.01$ while the factor ρ is chosen to 0.3. In order to ensure that the maximum computational time does not exceed its limit the total number of optimization iterations is fixed to four. The maximum number of iterations to find a feasible α is set to three as every iteration requires an additional integration of the system in order to evaluate the cost function $J_c(\triangle L_x^k + \alpha p^k)$. In this setup the total time for one optimization is less than 1.6 ms (for one direction).

C. Example optimization

This section presents details of an optimization example to illustrate the procedure. We are choosing a case in which the robot is walking in place while it is pushed by an external force. Fig. 5 shows the resulting footstep modification $\triangle L_x^i$ for 4 iterations and the evaluated cost function for these values. To visualize the Landscape of J_c, it is sampled for some discrete values of $\triangle L_x$. As one can see, the optimization is able to converge almost to the minimum of J_C in these iterations. The predicted evolution of the absolute inclination φ_x is shown in Fig. 6. As mentioned before, an initial integration is performed from t_0 to t_1 and the optimization considers the time interval $[t_1, t_2]$.

The progress of the initial costs and the corresponding optimized costs are shown in Fig. 7. In this example the robot is walking in place while it is pushed from behind with the force trajectory shown in Fig. 7. There are two characteristics observable: In the undisturbed case initial and optimized costs are close to zero, while the costs are significantly reduced by the optimization after the disturbance.

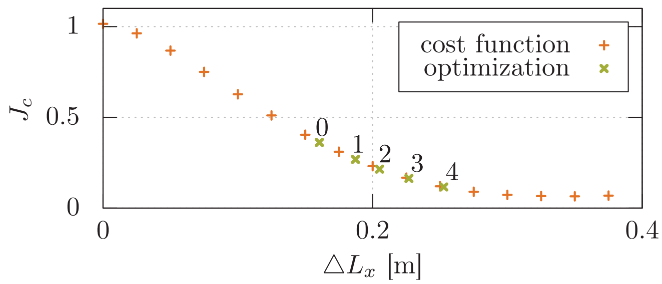


Fig. 5. Optimization results: Initial solution (0), and four iterations (1-4).

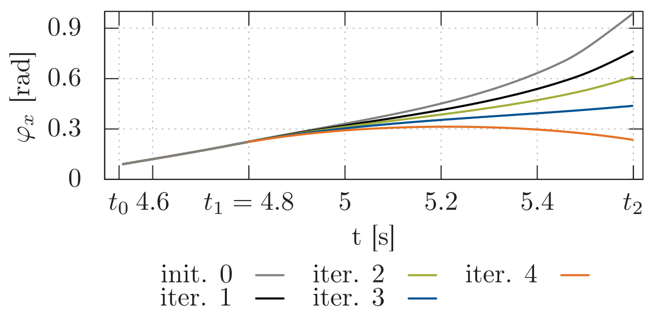


Fig. 6. Optimization results: predicted trajectory for the inclination φ_x for four optimization iterations.

VI. Model predictive trajectory modification

The proposed optimization of the step lengths has to be integrated into a framework, which continuously changes the ideal trajectories according to the computed step lengths. The algorithm is depicted in Fig. 8. The overall feedback input is the measured inclination φ_m and inclination rate $\dot{\varphi}_m$ from the inertial measurement unit (IMU) mounted to the robot's upper body. These quantities are fed into a state observer (see [28] for more details) using also measured contact forces and torques $\boldsymbol{\lambda}$ in order to filter unwanted fast motions (e.g. structural vibrations). The estimated values $\hat{\varphi}_0$, $\dot{\hat{\varphi}}_0$ are then used with the ideal task–space trajectories $\mathbf{w}_{id}$ for the initial integration until $t = t_1$. The predicted state $\hat{\mathbf{x}}(t_1)$ and the initial step length modification $\triangle L^0$ are used in the direct shooting method in order to calculate the optimal value $\triangle L$.

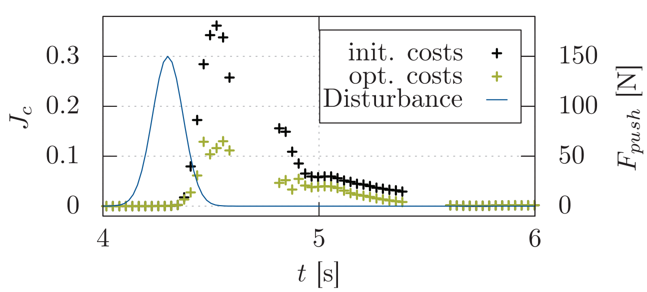


Fig. 7. Optimization results: initial and optimized costs for different moments after an external disturbance.

All computations are performed both for x– and y–direction with the corresponding variables, so indices indicating the direction are omitted.

The trajectory generation uses $\triangle L$ and the predicted inclination errors at the end of current step $\hat{\varphi}(t_1)$ to calculate stabilizing trajectory modifications $\triangle \mathbf{w}$. With the current state of the swing foot new trajectories are calculated with the set point $\triangle L$ at $t = t_1$ using fifth order polynomials. Since all trajectories are tracked in the upper body fixed FoR of the real robot, the predicted inclination has to be compensated in order to avoid early or late ground contact of the swing foot. The set point for the final angle is set to $-\hat{\varphi}_x(t_1)$ and $-\hat{\varphi}_y(t_1)$. The final step height modification uses the rotation matrix $\mathbf{A}_{IT}(-\hat{\varphi}(t_1))$ which transforms from the upper body fixed FoR to the inertial FoR with the overall step length

$$\triangle z = \mathbf{e}_z \mathbf{A}_{IT}(-\hat{\varphi}(t_1)) \begin{bmatrix} L_{x,id} + \triangle L_x \\ L_{y,id} + \triangle L_y \\ 0 \end{bmatrix}. \tag{10}$$

During the stance phase, the modified foot rotation is kept constant while the modified foot height is planned back to zero if $\triangle z < \triangle z_{min}$ and otherwise to half of its value. In order to increase the robustness the trajectory generation also uses an early contact event from ideal trajectory generation [2]. This event occurs when the swing foot hits the ground before the expected time. This might occur due to unrecognized uneven terrain or imperfect inclination compensation. Each foot is equipped with discrete contact switches (four in total per foot) which can be monitored in the trajectory generation module. If one or more contacts are closed, a transition to double support phase is initiated that causes a fast but continuous stop of all trajectories. Therefore all modification trajectories of the swing foot are planned to zero velocity as well in order to have a synchronized behavior.

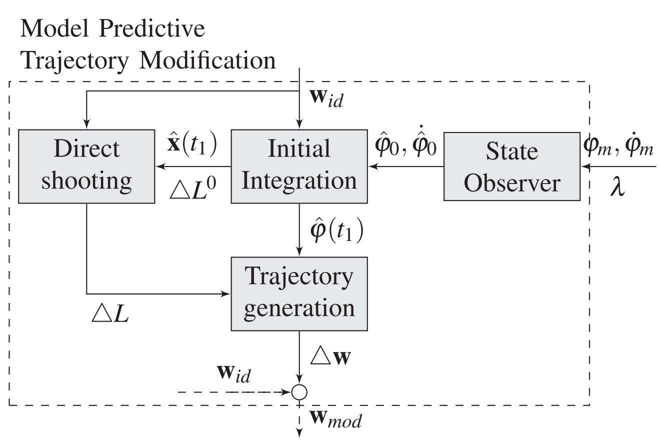


Fig. 8. Overview of the sensor feedback framework to trajectory generation.

VII. Experimental Results

The proposed method is implemented in the walking control system of the biped LOLA and tested in a simulation environment that considers unilateral and compliant contacts, motor dynamics as well as the joint control loop. The humanoid is commanded to walk with a step length of 0.4 m and a step time of 0.8 s forward for 4.5 m. At t=4.4 s we apply an external push ($F_x = 140\,\mathrm{N}, Fy = 80\,\mathrm{N}$ magnitude) to the robot. Fig. 9 shows the resulting inclination for the three cases

- off: no trajectory modification method,
- old: model predictive trajectory modification using our previously proposed method [1],
- new: proposed method in this paper.

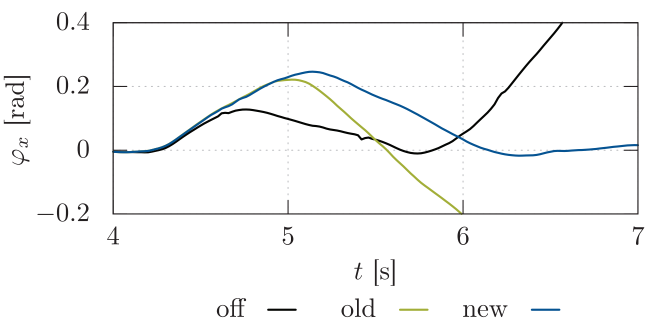


Fig. 9. Simulation results for no footstep modifications (off), our previously proposed method (old) and the new method (new) with an external push ($F_x = 140\,\mathrm{N}, Fy = 80\,\mathrm{N}$) during walking forward with 0.5 m/s.

As one can see, only the new method is able to stabilize the simulated robot for this disturbance. The old method uses only heuristic, which does not explicitly consider the influence of the footstep modification. Thus it is mandatory to include the influence of the modifications in a stabilization method when the robot is walking forward.

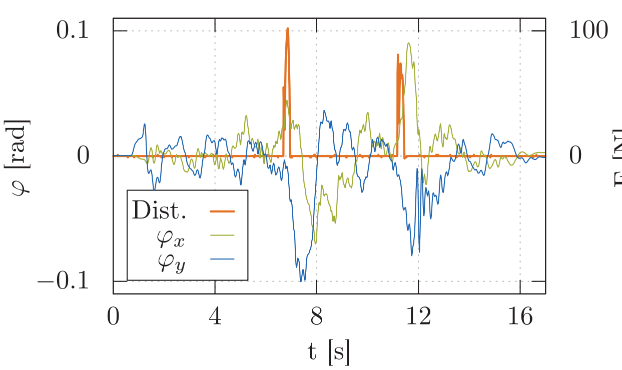


Fig. 10. Disturbance force and inclination errors while the robot is walking with approx. 0.5 m/s.

Finally, we tested our method in real–world experiments to show its efficiency and flexibility to different disturbances (varying magnitude, direction and timing). The presented example is performed with the same walking command as in the previously shown simulation. The robot is pushed twice during walking. The corresponding external forces are shown in Fig. 10. We integrated a force transducer to our pushing device in order the measure the transferred impulse. The ideal and the modified step length for this experiment are depicted

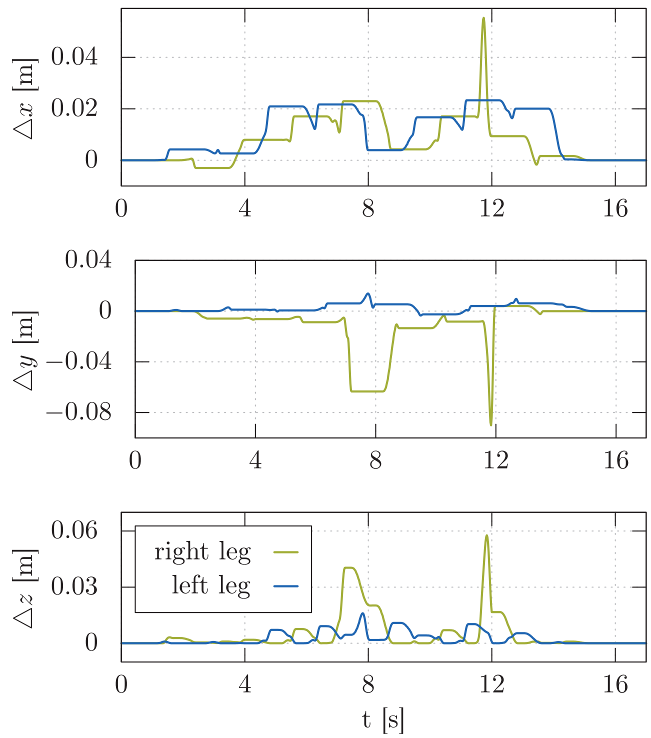


Fig. 11. Resulting trajectory modifications for the experiment.

in Fig. 12. The approximate moments when the pushes occur are visualized with red arrows. The corresponding trajectory modifications for the foot positions are shown in Fig. 11, modification of the orientation of the feet is omitted due to space limitations.

The video of the described experiment is attached to the present paper and photographs are shown in Fig. 13.

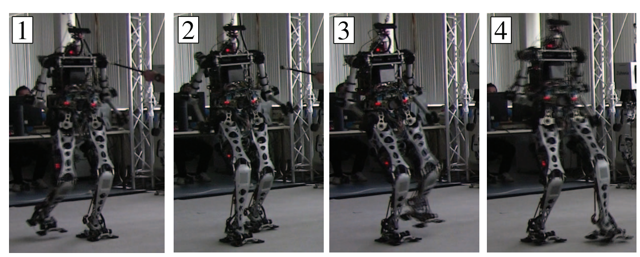


Fig. 13. Walking experiment. The robot is pushed in the first picture and stabilizes itself.

VIII. Conclusion

A new optimization based method to determine a stabilizing step length is presented in this work. A previously proposed nonlinear model of the biped robot is used in order to implement a model predictive optimization. The model considers ideal planned as well as modified trajectories which is essential for the usage in the optimization. The resulting problem is solved by direct shooting where terminal constraints are omitted in order to ensure computational times that allow an application in real–time. Further we present

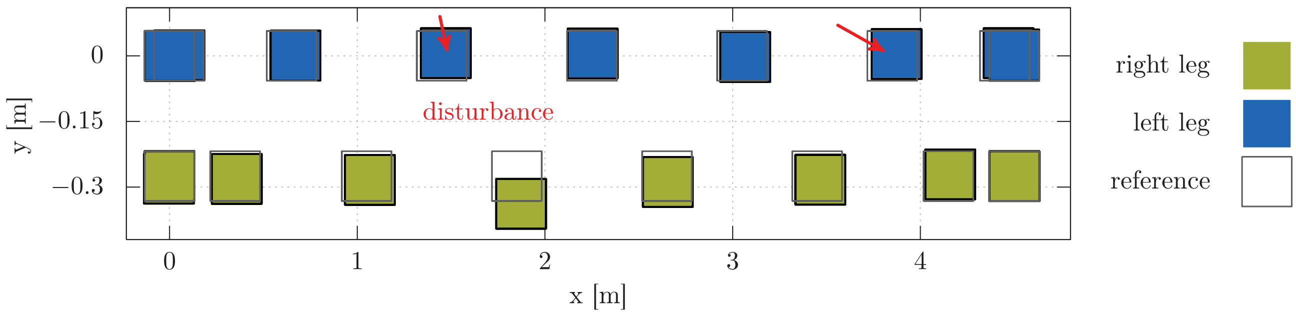


Fig. 12. Ideal and modified foot steps of the robot walking with approx. 0.5 m/s while it is pushed twice.

the integration of the method in the overall model predictive trajectory modification module and its implementation to the walking control system of our biped robot LOLA. With this method the robot is able to walk stably forward with approx. 0.5 m/s while it is pushed from arbitrary directions.

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 *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, no. Iros. Ieee, Sept. 2014, pp. 2572–2578.

[2] T. Buschmann, S. Lohmeier, and H. Ulbrich, "Biped walking control based on hybrid position/force control," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2009, pp. 3019–3024.

[3] Y. Fujimoto, S. Obata, and A. Kawamura, "Robust biped walking with active interaction control between foot and ground," *Proceedings. 1998 IEEE International Conference on Robotics and Automation*, vol. 3, no. May, pp. 2030–2035, 1998.

[4] K. Loffler, M. Gienger, and F. Pfeiffer, "Model based control of a biped robot," in *7th International Workshop on Advanced Motion Control. Proceedings*. IEEE, 2002, pp. 443–448.

[5] 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 *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2009, pp. 1601–1608.

[6] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara, and H. Hirukawa, "Biped Walking Pattern Generator allowing Auxiliary ZMP Control," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 2993–2999, Oct. 2006.

[7] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4489–4496, Oct. 2010.

[8] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *2014 IEEE International Conference on Robotics and Automation*. IEEE, May 2014, pp. 2589–2594.

[9] A. Sherikov, D. Dimitrov, and P.-b. Wieber, "Whole body motion controller with long-term balance constraints," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 444–450.

[10] 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.

[11] ——, "Sensor feedback modification methods that are suitable for the short cycle pattern generation of humanoid walking," *IEEE/RSJ International Conference on Intelligent Robots and System*, pp. 4214–4220, Oct. 2007.

[12] R. Tajima, D. Honda, and K. Suga, "Fast running experiments involving a humanoid robot," *2009 IEEE International Conference on Robotics and Automation*, pp. 1571–1576, May 2009.

[13] 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, June 1991.

[14] M. Missura and S. Behnke, "Online learning of foot placement for balanced bipedal walking," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 322–328.

[15] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pp. 200–207, Dec. 2006.

[16] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[17] G. Nelson, A. Saunders, N. Neville, B. Swilling, J. Bondaryk, D. Billings, C. Lee, R. Playter, and M. Raibert, "PETMAN: A Humanoid Robot for Testing Chemical Protective Clothing," *Journal of the Robotics Society of Japan*, vol. 30, no. 4, pp. 372–377, 2012.

[18] B. J. Stephens and C. G. Atkeson, "Push Recovery by stepping for humanoid robots with force controlled joints," *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pp. 52–59, Dec. 2010.

[19] J. Urata, K. Nishiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba, "Online decision of foot placement using singular LQ preview regulation," *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pp. 13–18, Oct. 2011.

[20] S. Lohmeier, "System design and control of anthropomorphic walking robot LOLA," *2009 IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 6, pp. 658–666, 2009.

[21] T. Buschmann, V. Favot, S. Lohmeier, M. Schwienbacher, and H. Ulbrich, "Experiments in fast biped walking," in *2011 IEEE International Conference on Mechatronics*. IEEE, Apr. 2011, pp. 863–868.

[22] H. Bock, M. Diehl, D. Leineweber, and J. Schlöder, "A direct multiple shooting method for real-time optimization of nonlinear DAE processes," *Progress in Systems and Control Theory*, vol. 26, pp. 245–267, 2000.

[23] M. Diehl, H. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear MPC and moving horizon estimation," *Nonlinear Model Predictive Control*, pp. 391–417, 2009.

[24] 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.

[25] J. T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *Survey of numerical methods for trajectory optimization*, vol. 21, no. 2, pp. 193–207, 1998.

[26] T. Matsumoto, T. Takenaka, and T. Yoshiike, "Gait generation device for legged mobile robot," 2004.

[27] J. Nocedal and S. J. Wright, "Numerical Optimization," pp. 1–651, 2004.

[28] R. Wittmann, A.-C. Hildebrandt, D. Wahrmann, T. Buschmann, and D. Rixen, "State Estimation for Biped Robots Using Multibody Dynamics (submitted to IROS)," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.