Proceedings Paper:
You, Y, Zhou, C orcid.org/0000-0002-6677-0855, Li, Z et al. (1 more author) (2017) A study of nonlinear forward models for dynamic walking. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). 2017 IEEE ICRA, 29 May - 03 Jun 2017, Singapore. IEEE , pp. 3491-3496. ISBN 978-1-5090-4633-1

https://doi.org/10.1109/ICRA.2017.7989399

A Study of Nonlinear Forward Models for Dynamic Walking

Yangwei You*, Chengxu Zhou*, Zhibin Li**, Nikos Tsagarakis*

Abstract

This paper offers a novel insight of using nonlinear models for the control to produce more robust and natural walking gaits for humanoid robots. The sagittal and lateral gait control need to be treated differently, and hence we proposed two types of suitable nonlinear models, which allow the forward simulations to look ahead and thus predict accurately the future trajectory/state at the end of the current step. Subsequently, by performing multiple forward simulations in a similar manner for the next step and using gradient descent method, an appropriate foot placement can be found to achieve precise walking speed. By doing this two-step lookahead, all the trajectories of support and swing leg can be determined. Our proposed controller can plan trajectories at the beginning of each step or actively re-plan according to task state errors. It is validated effectively in simulation studies performed in both ADAMS and Open Dynamic Engine. The robot can successfully traverse up/down a stair and recover from pushes with more natural looking gaits compared to conventional bent knee style. The reasonable computational time from this study also indicates feasible real-time implementation on real robots.

I. Introduction

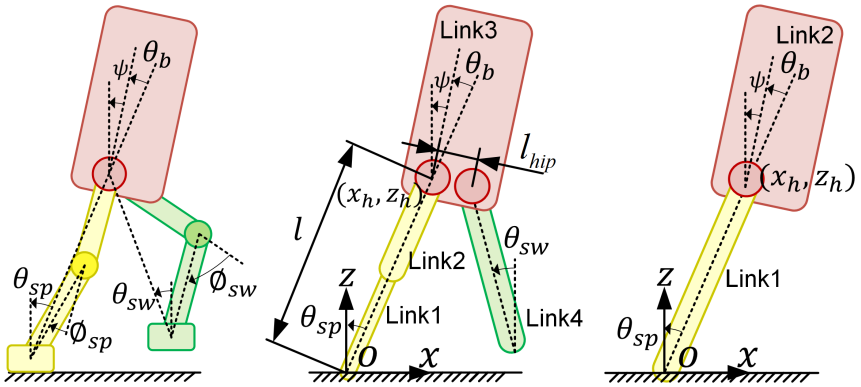
The high versatility and adaptability of animals on rough terrains motivates a lot of research developments on legged robots which are expected to walk and run in a natural environment [1]–[4]. One of the most popular models for walking is the linear inverted pendulum model (LIPM) [5], which can provide analytic solution and therefore needs very little computational resource. The cart-table model [6] is also often used and usually integrated with the preview control to manipulate the zero moment point (ZMP) inside the contact polygon. It can make full use of the ankle torque to achieve better performance than the LIPM. The linear inverted pendulum plus flywheel model has also been used for bipedal walking to consider the upper body movements [7]. However, all of them apply linear models, which inevitably constrain the robot's potential of physical achievable locomotion and also make its movement unnatural. Bent knee of this kind of robot is an obvious evidence, which is unnatural comparing with human walking and requires more torque in the knee joints resulting more energy consuming. To overcome such problems and produce more versatile gaits, we propose a novel control method that explores nonlinear models.

Biological research revealed that the movement of legged animals can be represented better by a spring loaded inverted pendulum (SLIP) model [8]. And based on this idea, Raibert developed several successful legged robots from a one-legged hopper to a quadruped [9]. He used foot placement to control the forward velocity of these robots by a very simple equation. This method has been also successfully applied to many other robots [10], [11]. As the improvements of this method, the tabular control [12] and the approximate optimal control [13] are proposed. However, both approaches needed lots of experimental data and were performed off-line. Recently, Dai *et al*. transformed the whole body motion planning of humanoid robot to a nonlinear optimization problem and obtained impressive results where the robot can complete really complex tasks such as passing monkey bars [14]. Nevertheless, this method is quite time-consuming and needs to plan the trajectories off-line as well.

Studies on sensorimotor control reveals that the central nervous system (CNS) internally simulates forward the behaviors of the motor system for planning and control, thus "forward" model is termed (or internal model) as a dynamic representation for the motor system to use the current state to predict the future state [15]. Also, forward/internal models inherit the delay property coming from sensory feedback and motor response [16], which can be considered a-priori in the control stage while generating a new motor command [17]. New study also proves that not only the vertebrates, but also the invertebrates, the Plathemis lydia, whose sensorimotor control exploits the dynamic models of both itself and the prey for the prediction and planning during the high-performance control of interception steering [18].

While some recent robotics research have used inverse dynamics approach [19], similar to the inverse models for CNS, to solve whole-body control problem, we are motivated to investigate what types of models can be used, as those forward models in CNS, and how to exploit forward models for planning and control that can further enhance dynamic walking in terms of robustness and human-level of likeness. Particularly, inspired by Raibert's idea of controlling foot placement to balance legged robot, we hereby proposed two nonlinear models for forward simulation to decide foot placement more precisely by combining multiple iterations of gradient descent search. We will show that these nonlinear models have much better representation of the full robot dynamics than the over simplified single point mass model and meanwhile is computationally easy to realize. Therefore, more accurate foot placements can be achieved based on these improved models compared to the previous trial-and-error tuning of control coefficients as in [10]. Our study of nonlinear models offers new locomotion features as follows:

*Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy.
Tel.: +39-010-71781407. Fax: +39-010-71781232. {yangwei.you, chengxu.zhou, nikos.tsagarakis}@iit.it
**School of Informatics, University of Edinburgh, 10 Crichton St, Edinburgh EH8 9AB, United Kingdom. Email: zhibin.li@ed.ac.uk



(a) Full model (b) Four-link model (c) Two-link model

Fig. 1. Two nonlinear models derived from a full robot model.

1) The nonlinear models take the dynamics of support and swing legs into account, i.e. distributed mass and inertia tensor of the rigid bodies;
2) The kinematics of the legs are included in the model so as to resolve the knee singularity issue;
3) Constraint on center of mass (COM) height for linearization is no longer required, which allows the generation of a more diverse COM motion and results in natural looking walking;
4) The change of mechanical energy due to ground impacts is included;
5) The actuator dynamics can be incorporated into the forward simulation.

Our paper is organized as follows. In Section II, two nonlinear models are introduced, and the mapping between a real robot and these models is also explained. Section III presents the algorithm of using forward simulation of nonlinear models to predict the foot placement, and the trajectory planning that is consistent with the newly updated foot placement. In Section IV, several simulations are performed to demonstrate the robust and human-like performance of the proposed method. The paper ends with conclusions and an outlook for future research.

II. Nonlinear Models

A. Nonlinear Model for Sagittal Dynamics

To capture major dynamic characteristics of bipedal walking in sagittal/lateral plane, the robot is simplified to a four-link model as Fig. 1 (b) shows. In this four-link model, the support leg and the swing leg are modeled as a prismatic link and a constant link respectively. Here, using a prismatic joint to represent the revolute knee joint is for improving the stability of numerical calculation. As for the swing leg, the dynamic property of the swing leg caused by the changing leg length is much less critical, rather, the dynamics of swing motion matters. Hence, it is reasonable to model the swing leg as one link with constant length. The foot is modeled as force wrench by applying torque to first joint. Assume no slippage, the equation of motion can be written as a fix-based robot instead of a floating base one for faster computation:

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

where $\mathbf{q}$ is the four degrees of freedom, $\mathbf{M}(\mathbf{q})$ is inertia matrix, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ is the sum of centrifugal, Coriolis and gravity forces, and $\mathbf{F}$ is generalized joint torque or force.

We are concerned about the task space vector $\mathbf{s} = [\theta_{\text{sp}}\ l\ \psi\ \theta_{\text{sw}}]^T$ denoted in Fig. 1. We define the line connecting hip and ankle joints as the virtual leg. Then θ_{sp} is the angle between virtual support leg and vertical line, l is the length of virtual support leg, ψ is the upper body posture, and θ_{sw} is the angle between virtual swing leg and vertical line. Thus, we can rewrite the dynamic model as:

$$\mathbf{M}^*(\mathbf{s})\ddot{\mathbf{s}} + \mathbf{h}^*(\mathbf{s}, \dot{\mathbf{s}}) = \mathbf{F}^*. \tag{2}$$

The task space vector $\mathbf{s}$ can be transformed to $\mathbf{q}$ according to the robot's geometric structure. Equation (2) provides options to include either closed loop control, actuator dynamics, or both into forward simulation. The torques or forces calculated from close loop control and actuator dynamics can be imported to $\mathbf{F}$ and then the forward simulation can realized via numerical integration.

B. Nonlinear Model for Lateral Dynamics

It shall be noted that the model in Fig. 1 (b) and the method above can be perfectly applied to both sagittal and lateral gait control. The only difference is the parameter l_{hip} that describes the separation of hips. However, interestingly, our prior

comparison study found out that the lateral dynamics has a particular feature which allows the model to be further simplified, as depicted by a two-link model in Fig. 1 (c), where the first link represents the support leg and the second link is one equivalent body lumping the upper body and the swing leg.

For the sagittal gait, the target is to pass over the support leg and continue to place next foot, so it includes a convergent phase and a divergent phase. The change of support leg length affects the potential energy of the whole robot and the future evolution of robot state. In contrast, the lateral gait needs to achieve a stable upper body oscillation laterally, and the upper body never surpasses the support foot sideways. Therefore, a limited variation of support leg length will not cause upper body to overshoot sideway. Meanwhile, the swing leg has much smaller range of motion laterally than sagittally, thereby the torso dynamics dominates over swing leg dynamics in the lateral scenario, which leads to no distinctive difference when a two-link model is used.

Our control tries to keep the knee as straight as possible for lower energetic cost, similar to what humans do. So the support link in Fig. 1 (c) is of full leg length. Therefore, in the lateral plane, the task space vector $\mathbf{s}$ in dynamic model 2 only contains θ_{sp} and ψ.

Given the expected posture ψ of torso with perfect tracking and ankle torque of support leg, we can get the trajectory of θ_{sp} without considering internal hip joint torque:

$$\ddot{\theta}_{\text{sp}} = f(\tau_1, \ddot{\psi}, \theta_{\text{sp}}, \psi, \dot{\theta}_{\text{sp}}, \dot{\psi}). \tag{3}$$

The same knack can also be applied to systems with four links or even more links. Similar arguments about this kind of simplification were made via conservation of angular momentum in [20].

C. From Nonlinear Models to the Robot

Here we address the mapping between model and robot concerning the support leg only, and the planning of swing leg trajectory will be explained in the Section III-B. In the four-link model, it is easy to map kinematically due to the same degrees of freedom. As for the two-link model, the distance from hip to ankle joint in the robot is always changing while the length of link 1 in the two-link model is constant. Define the hip position and velocity from the state estimation as (x_h, z_h) and $(\dot{x}_h, \dot{z}_h)$ respectively, as shown in Fig. 1, θ_{sp} and $\dot{\theta}_{\text{sp}}$ are calculated without considering the movement along virtual leg direction as:

$$\begin{aligned} \theta_{\text{sp}} &= \arctan(-x_h/z_h), \\ \dot{\theta}_{\text{sp}} &= -\dot{x}_h \cos(\theta_{\text{sp}}) - \dot{z}_h \sin(\theta_{\text{sp}}). \end{aligned} \tag{4}$$

III. Forward Simulation for Control

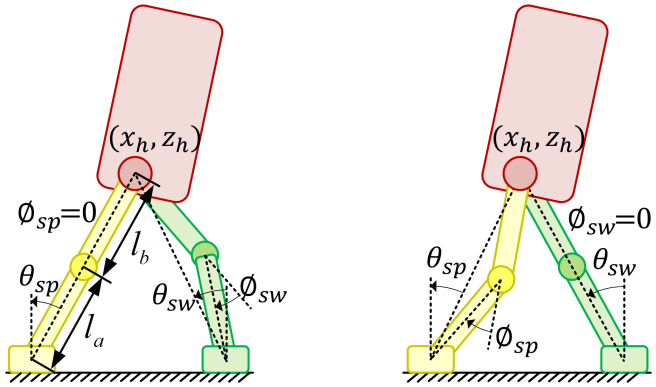
For a simple linear model, e.g the cart-table model, to achieve the optimal ZMP trajectory, a model preview control (MPC) can be easily implemented. However, once the model is nonlinear and particularly involves discrete change of energy state as in our models, to apply MPC requires a lot of computation and real-time requirement is hard to guarantee. Instead of optimizing the states of each sampling time, one option to significantly reduce the computation is to optimize only one key state. A great example is Raibert's work in [9] that a stable walking was realized by using an intuitive equation to determine the foot placement of touch-down state. One disadvantages is that the control coefficients are empirically tuned due to the unmodeled dynamics. On the contrary, our proposed nonlinear models are more accurate so as to avoid this tuning problem.

A. Accurate Foot Placement Control

Foot placement primarily determines the discrete nature of step-to-step transition, and thus the stability of walking and control of the robot movement [9], [21], [22]. As [22] said, the foot placement has an almost linear relationship between the velocity of robot. Hence given the robot state predicted by the forward simulation at the end of the current step, it should be very easy to use the gradient descent method to find an appropriate foot placement for the next step to reach a desired state. This is the essential idea of utilizing forward simulation for accurate foot placement control in the presence of multi-body effects during very dynamic locomotion.

Here we define foot placement as the initial angle of support leg $\theta_{\text{sp}}^{\text{ini}}$ at touch down. To determine the foot placement for next step, we need first to obtain the initial velocity ${}^{\text{k+1}}\dot{\theta}_{\text{sp}}^{\text{ini}}$ of next step. "k" and "k+1" in the left superscript mean current and next step separately, and the right superscripts "ini", "end" dedicate the starting and ending moment of one step. ${}^{\text{k+1}}\dot{\theta}_{\text{sp}}^{\text{ini}}$ can be derived from the predicted state $({}^{\text{k}}\theta_{\text{sp}}^{\text{end}}, {}^{\text{k}}\dot{\theta}_{\text{sp}}^{\text{end}})$ of current step. Fig. 2 shows the robot configurations at touch-down moment when the step is switched. To represent the ground impact, the energy along the direction of virtual support leg of next step is lost, and the rest produces only the angular velocity around the new stance foot. Then ${}^{\text{k+1}}\dot{\theta}_{\text{sp}}^{\text{ini}}$ can be calculated as in (5). For narrative convenience, we term the switch function for both two- and four-link models as $f_{\text{sw}}({}^{\text{k}}\theta_{\text{sp}}^{\text{init}}, {}^{\text{k+1}}\theta_{\text{sp}}^{\text{ini}}, {}^{\text{k}}\dot{\theta}_{\text{sp}}^{\text{end}})$,

$$\begin{aligned} {}^{\text{k+1}}\dot{\theta}_{\text{sp}}^{\text{ini}} &= f_{\text{sw}}({}^{\text{k}}\theta_{\text{sp}}^{\text{init}}, {}^{\text{k+1}}\theta_{\text{sp}}^{\text{ini}}, {}^{\text{k}}\dot{\theta}_{\text{sp}}^{\text{end}}) \\ &= {}^{\text{k}}\dot{\theta}_{\text{sp}}^{\text{end}} \cos({}^{\text{k+1}}\theta_{\text{sp}}^{\text{ini}} - {}^{\text{k}}\theta_{\text{sp}}^{\text{end}}). \end{aligned} \tag{5}$$



(a) Straight support leg (b) Straight swing leg

Fig. 2. Two robot configurations for touch-down.

Algorithm 1: Determine foot placement (Pseudo code).

Input : ${}^{k}\theta_{sp}^{ini}, {}^{k}\dot{\theta}_{sp}^{ini}, {}^{k+1}\dot{\theta}_{sp}^{des}$.
Output: ${}^{k+1}\theta_{sp}^{ini}$

```
k θ_sp^end, k θ̇_sp^end ← ForwardSim(k θ_sp^ini, k θ̇_sp^ini)
k+1 θ_sp^ini(0) = 0
k+1 θ_sp^ini(1) = α
for n ← 1 to N do
    k+1 θ̇_sp^ini(n−1) ← f_sw(k θ_sp^end, k+1 θ_sp^ini(n−1), k θ̇_sp^end)
    k+1 θ̇_sp^ini(n) ← f_sw(k θ_sp^end, k+1 θ_sp^ini(n), k θ̇_sp^end)
    k+1 θ̇_sp^end(n−1) ←ForwardSim(k+1 θ_sp^ini(n−1), k+1 θ̇_sp^ini(n−1))
    k+1 θ̇_sp^end(n) ← ForwardSim(k+1 θ_sp^ini(n), k+1 θ̇_sp^ini(n))
    if |(k+1 θ̇_sp^end(n) − k+1 θ̇_sp^des)| < ε or n == N then
        return k+1 θ_sp^ini(n)
    else
        Δ k+1 θ_sp^ini = ((k+1 θ̇_sp^end(n) − k+1 θ̇_sp^des))(k+1 θ_sp^ini(n) − k+1 θ_sp^ini(n−1)) / (k+1 θ̇_sp^end(n) − k+1 θ̇_sp^end(n−1))
        k+1 θ_sp^ini(n+1) = k+1 θ_sp^ini(n) − Δ k+1 θ_sp^ini
    end
end
```

Algorithm 1 is used to determine the foot placement. First we predict the robot state $({}^{k}\theta_{sp}^{end}, {}^{k}\dot{\theta}_{sp}^{end})$ at the end of current step by performing forward simulation from the current measured state. Then ${}^{k+1}\theta_{sp}^{ini}(0)$ and ${}^{k+1}\theta_{sp}^{ini}(1)$ are initial values to start the iteration of gradient descent method. Afterwards ${}^{k+1}\dot{\theta}_{sp}^{ini}(n-1)$ and ${}^{k+1}\dot{\theta}_{sp}^{ini}(n)$ are derived from the corresponding switch function (5). Based on this, ${}^{k+1}\dot{\theta}_{sp}^{end}(n-1)$ and ${}^{k+1}\dot{\theta}_{sp}^{end}(n)$ are calculated from forward simulation using (2). Then, if the error is smaller than the tolerance ε or the iteration times are more than the maximum number, the result ${}^{k+1}\theta_{sp}^{ini}(n)$ is returned; otherwise, a gradient descent method will be applied to calculate ${}^{k+1}\theta_{sp}^{ini}(n+1)$ for the next iteration.

B. Mapping From Virtual Legs to Revolute Knees

From the algorithms mentioned above, we can obtain the trajectory of θ_{sp} of support leg and $(\theta_{sw}^{end}, \dot{\theta}_{sw}^{end})$ of swing leg. By now, we still don't take the knee joint state $(\phi_{sp}, \dot{\phi}_{sp})$ and $(\phi_{sw}, \dot{\phi}_{sw})$ into account. As illustrated by two robot configurations at touch-down in Fig. 2, our control keeps the support leg straight during the most period of walking. In this case, the knee angle ϕ_{sp}^{end} and ϕ_{sw}^{end} of support and swing leg at touch-down can be calculated given the lengths l_a and l_b of shin and thigh:

$$\phi_{sp}^{end} = \begin{cases} \pi - \arccos(\frac{l_a^2 + l_b^2 - c_{sp}^2}{2 l_a l_b}), & \theta_{sp}^{end} < \theta_{sw}^{end} \\ 0, & \theta_{sp}^{end} \geq \theta_{sw}^{end} \end{cases}$$

$$\phi_{sw}^{end} = \begin{cases} 0, & \theta_{sp}^{end} < \theta_{sw}^{end} \\ \pi - \arccos(\frac{l_a^2 + l_b^2 - c_{sw}^2}{2 l_a l_b}), & \theta_{sp}^{end} \geq \theta_{sw}^{end} \end{cases}$$

$$c_{sp} = (l_a + l_b)\frac{\cos(\theta_{sw}^{end})}{\cos(\theta_{sp}^{end})}, c_{sw} = (l_a + l_b)\frac{\cos(\theta_{sp}^{end})}{\cos(\theta_{sw}^{end})}.$$

With an expected upright posture of upper body $\psi = 0$, then by now we can get the measured current state $(\phi_{\text{sp}}^{\text{init}}, \psi^{\text{init}}, \theta_{\text{sw}}^{\text{init}}, \phi_{\text{sw}}^{\text{init}}, \dot{\phi}_{\text{sp}}^{\text{init}}, \dot{\psi}^{\text{init}}, \dot{\theta}$ and the expected end state $(\phi_{\text{sp}}^{\text{end}}, \psi^{\text{end}}, \theta_{\text{sw}}^{\text{end}}, \phi_{\text{sw}}^{\text{end}}, \dot{\phi}_{\text{sp}}^{\text{end}}, \dot{\psi}^{\text{end}}, \dot{\theta}_{\text{sw}}^{\text{end}}, \dot{\phi}_{\text{sw}}^{\text{end}})$ of current step, so we can smoothly connect them by polynomials and keep continuity of the state. An intermediate point should be added into the trajectory of ϕ_{sw} to lift up the swing leg and create foot-ground clearance for walking. Combining them with the trajectory of θ_{sp} generated by forward simulation and the desired $\psi = 0$, we can obtain the trajectories for all joints. As we mentioned before, this method can be applied to the movement of both sagittal and lateral planes, the only difference is the parameters of model.

IV. Simulation

To evaluate the effectiveness and performance, our proposed control method was tested by several different simulation scenarios in two simulation environments. Both of them showed promising results.

A. Traversing a Stair Blindly

This simulation was performed on a planar bipedal robot built in ADAMS.The mass of torso, thigh, shin and sole are 19.8, 2.8, 2.5 and 0.35 kg respectively, while their moments of inertia are 0.3, 0.02, 0.02, 0.001 kgm2, and the limb dimension of 0.4, 0.25, 0.25, 0.1 m. The actuators in ankle and knee joints produced torque to track expected joint positions with stiffness 1000 N $\cdot$ m/rad and viscous damping 30 N $\cdot$ m $\cdot$ s/rad while the hip joints implemented PD controllers to track task space targets, upper body posture for support leg and foot placement for swing leg. The proportional and derivative coefficients are 500 N $\cdot$ m/rad and 20 N $\cdot$ m $\cdot$ s/rad individually. In this simulation, we used the four-link model with joint actuator dynamics included and trajectories were generated via our controller at the beginning of each step. The maximum iteration times N was 10, the tolerance ε for terminating the iteration of forward simulation was 0.05 rad/s, the period of one walking step was 0.4 s and the time step for internal forward simulation was 0.01 s. The robot successfully walked through a stair of 5 cm height blindly in the sagittal plane. The movement was natural and compliant as the snapshot Fig. 3 shows. The angular velocity $\dot{\theta}_{\text{sp}}$ of virtual support leg and upper body posture ψ are given in Fig. 4 and 5 separately, where the solid blue line is the measured data, and the red dash-dot line is the trajectory planned on-line for each step.

The Fig. 4 shows that the robot stepped up and down the stair at about 6 s and 9 s. When the robot was stepping up, the sole touched the ground earlier than expected, so the planning of next step was triggered immediately. Because of the limited torque at ankle joint and early landing, the angular velocity $\dot{\theta}_{\text{sp}}$ of following step was not tracked very well. However, for the next second step, a good foot placement was computed by our controller based on nonlinear models, and the robot recovered to a stable state very quickly. Similar situation happened during the step-down in which the landing was a bit later than expected. Nevertheless, the robot can resist the ground disturbance quit well.

Fig. 5 demonstrates a good tracking of upper body posture ψ. Even when stepping up and down, the forward model still can generate a feasible posture trajectory for the robot to track accurately. This simulation indicates that the controller built on the forward model is very robust and can help the robot recover to desired state within a few steps.

B. Sagittal Push Recovery

To further study the capability of disturbance rejection, a forward push recovery simulation is carried out on the same platform as the previous one. In this simulation, a push force of 100 N is applied in the center of upper body starting from 2 s and lasted for 0.1 s as shown in Fig. 6. The simulation result is shown in Fig. 7 and 8. The definition of data lines in these figures is also the same as previous simulation.

When applying an external force, a large error of $\dot{\theta}_{\text{sp}}$ was invoked immediately, because it was not modeled in the controller. During the next step after the push, by replanting the movement via our proposed nonlinear model, the controller can restore the nominal gait by choosing a correct foot placement. As the previous simulation, the upper body posture was tracked strictly.

C. Lateral Push Recovery

The simulations above only concern about the movement in the sagittal plane. Here, the Open Dynamics Engine (ODE) was used to validate the proposed controller's effectiveness in the lateral direction on a simulated humanoid robot developed in [23]. This robot has the same joint configuration and kinematics as the COMAN robot [24]. All the joints are set to position control. According to our experience, the simulation results match quite well with experiment data.

The robot walks in a 3-dimensional ODE simulation, meaning that the robot has no constraints in the sagittal plane. Therefore, a simple trajectory generator was implemented to steer the ankle for keeping upper body in the same plane and orienting the swing foot to be parallel with the ground. In the lateral plane, the proposed control method was applied by using the two-link variant as the forward model and the trajectories were generated whenever the tracking errors of θ_{sp} and $\dot{\theta}_{\text{sp}}$ exceeded limits of 0.1 rad and 1 rad/s or the planned trajectories were expired. The maximum iteration number N was 5, the tolerance ε was 0.02 rad/s, the duration of one walking step is 0.4 s, and the time step for interal forward simulation is 0.002 s. Firstly, a push force of 3000 N was applied in the center of upper body for 0.01s as shown in Fig. 9. The angular velocities $\dot{\theta}$ of virtual left and right legs are shown in Fig. 10 and 11.

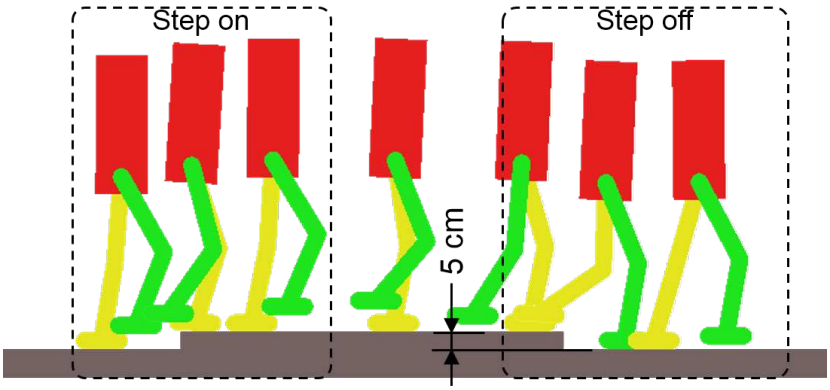


Fig. 3. Snapshots of crossing a stair.

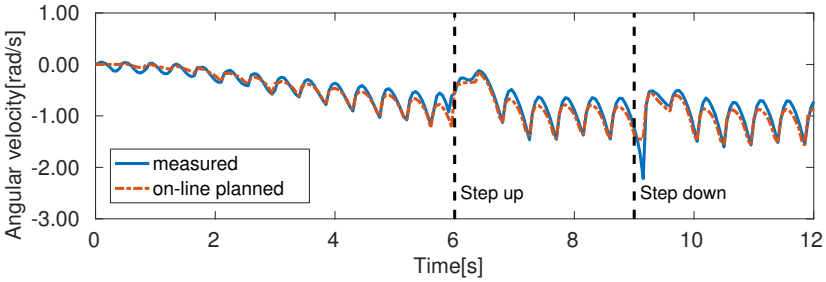


Fig. 4. Angular velocity of virtual support leg when crossing a step.

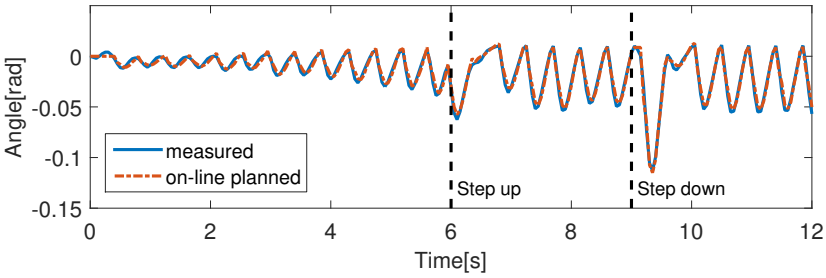


Fig. 5. Upper body posture when crossing a step.

In Fig. 9 to 11, the robot was pushed at about 1.5 s, and a big tracking error evoked trajectory replanning. The replanned trajectories were able to guide the robot to stabilize after two steps. The tracking error was quite small when the robot was stepping in place stably.

Fig. 13 shows an additional aggressive test during which the robot was intermittently impacted three times by a ball with mass of 1 kg and initial velocity of 10 m/s. Nevertheless, the robot was able to sprawl legs aside and position the foot swiftly to stop falling over. The high robustness of this controller was showcased by these trials. Note that in all these simulations, the robot had very straight knee feature which was comparable to humans. Since the forward model already takes the leg kinematics into consideration, the knee singularity issue is avoided in essence.

The coded algorithm in C++ runs each query of the iterative planning for less than 0.5 ms, which is very efficient as the computational time shown in Fig. 12. This is entirely feasible for real-time implementation and permits our future hardware validation.

V. Conclusion and Future Work

This study explores the forward model concept inspired by the sensorimotor control realm and applies the principle to the control of robust and dynamic bipedal walking. We first studied good candidates of nonlinear models for forward simulation, four-link and two-link models, which are suited to meet different requirements of sagittal and lateral gaits towards the minimum computation without downgrading much of the accuracy. These models can also include the characteristics of the control system, i.e. closed loop control and actuator dynamics. Based on these nonlinear models, we propose an iterative algorithm to best exploit forward simulation and automatically search for precise foot placement. With the proposed method, a planar bipedal robot was able to blindly travel over a stair and recover from a push successfully in simulation. Furthermore, the simulated COMAN robot was able to withstand several aggressive impact attacks without falling over.

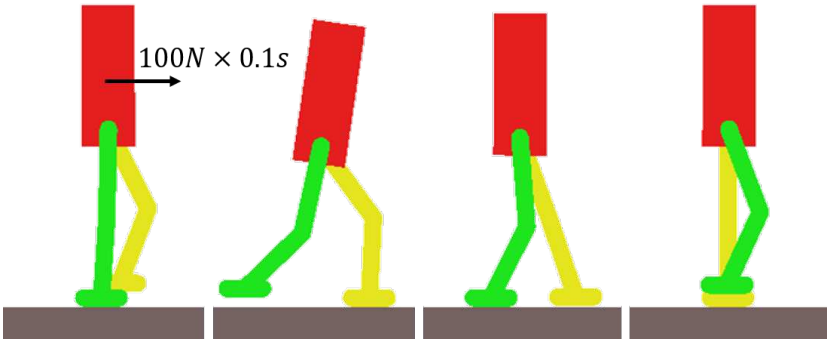


Fig. 6. Snapshots of push recovery.

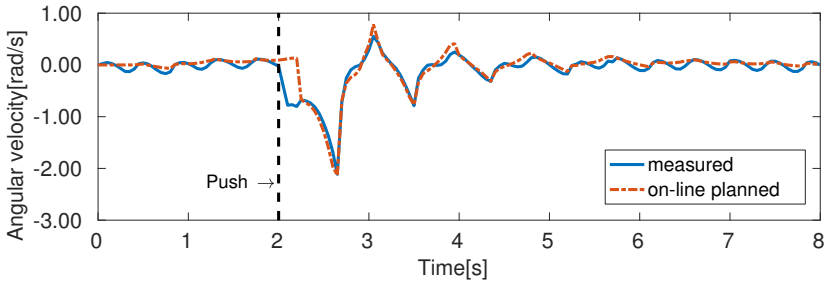


Fig. 7. Angular velocity of virtual support leg in push recovery.

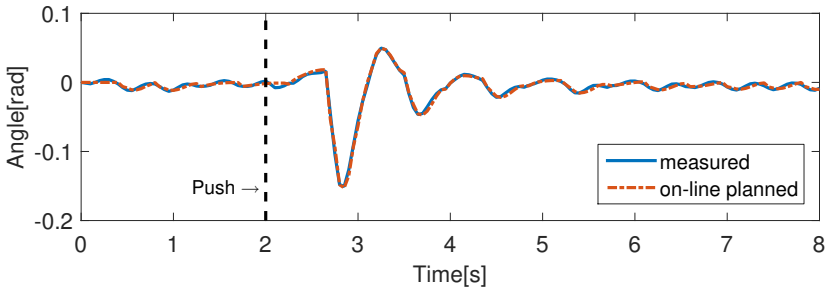


Fig. 8. Upper body posture in push recovery

This initial study separately validated the algorithms in two planar cases under different simulation software and programming languages, so the future work will port the code and integrate them into one codebase for the control of 3D walking. Although the control strategy now only works on these two proposed nonlinear models, other models can also be embedded inside to accomplish other tasks, e.g. a SLIP model for running. A more challenging research we shall quest for is to combine our forward model approach together with the state of the art inverse dynamics method, which overall will be very comparable to how the cerebellum produces the motor control [17].

ACKNOWLEDGMENT

This work is supported by the European Horizon 2020 robotics program CogIMon (ICT-23-2014 under grant agreement 644727) and FP7 European project WALK-MAN (ICT 2013-10).

REFERENCES

[1] M. Ahmadi and M. Buehler, "Stable control of a simulated one-legged running robot with hip and leg compliance," *Transactions on Robotics and Automation*, vol. 13, no. 1, pp. 96–104, 1997.

[2] J. Pratt and G. Pratt, "Intuitive control of a planar bipedal walking robot," in *International Conference on Robotics and Automation*, vol. 3, 1998, pp. 2014–2021.

[3] M. Hutter, C. D. Remy, M. A. Hoepflinger, and R. Siegwart, "Scarleth: Design and control of a planar running robot," in *International Conference on Intelligent Robots and Systems*, 2011, pp. 562–567.

[4] C. Zhou, X. Wang, Z. Li, and N. Tsagarakis, "Overview of Gait Synthesis for the Humanoid COMAN," *Journal of Bionic Engineering*, vol. 14, no. 1, pp. 15–25, 2017.

[5] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, "A realtime pattern generator for biped walking," in *International Conference on Robotics and Automation*, vol. 1, 2002, pp. 31–37.

[6] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *International Conference on Robotics and Automation*, vol. 2, 2003, pp. 1620–1626.

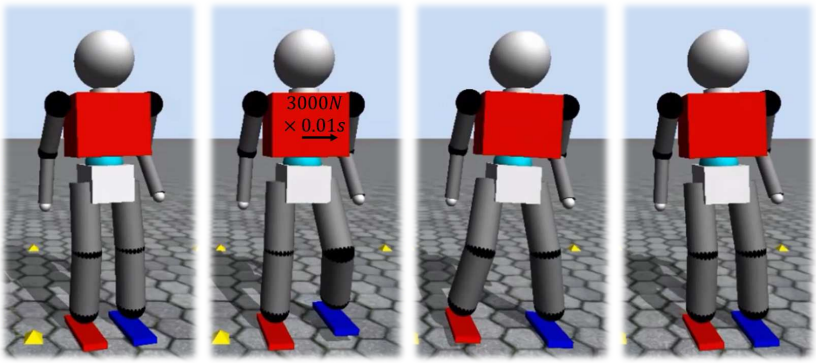


Fig. 9. COMAN pushed by a lateral force.

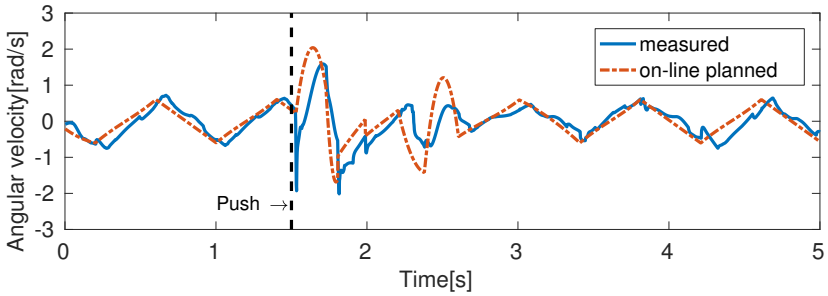


Fig. 10. Angular velocity of left virtual leg.

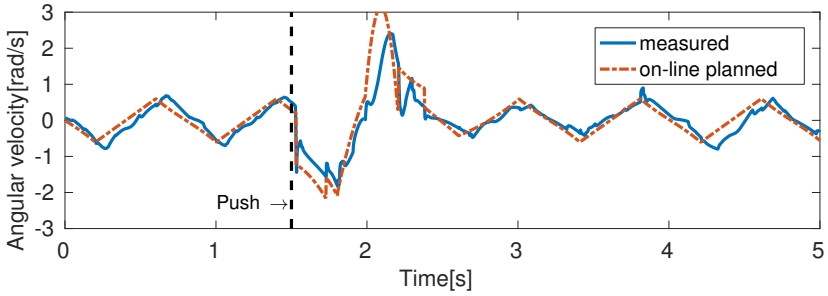


Fig. 11. Angular velocity of right virtual leg.

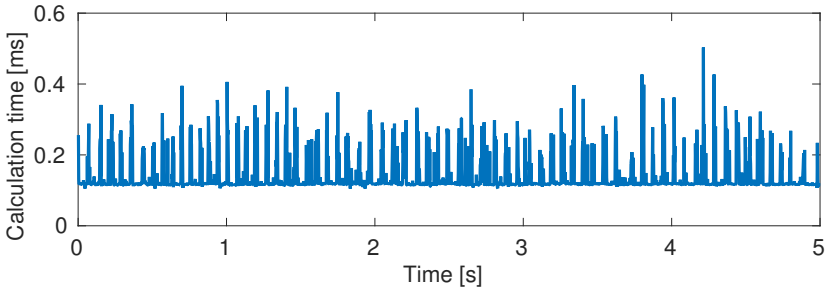


Fig. 12. Time consuming of controller.

[7] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *International Conference on Humanoid Robots*, 2006, pp. 200–207.
[8] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behaviour explains basic dynamics of walking and running," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 273, no. 1603, pp. 2861–2867, 2006.
[9] M. H. Raibert *et al.*, *Legged robots that balance*. MIT press Cambridge, MA, 1986, vol. 3.
[10] M. Raibert, K. Blankespoor, G. Nelson, R. Playter *et al.*, "Bigdog, the rough-terrain quadruped robot," in *Proceedings of the 17th World Congress*, vol. 17, no. 1, 2008, pp. 10 822–10 825.
[11] D. W. Marhefka, D. E. Orin, J. P. Schmiedeler, and K. J. Waldron, "Intelligent control of quadruped gallops," *Transactions on Mechatronics*, vol. 8, no. 4, pp. 446–456, 2003.
[12] M. H. Raibert and F. C. Wimberly, "Tabular control of balance in a dynamic legged system," *Transactions on Systems, Man and Cybernetics*, no. 2, pp. 334–339, 1984.
[13] R. L. Tedrake, "Applied optimal control for dynamically stable legged locomotion," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.
[14] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with simple dynamics and full kinematics," DTIC Document, Tech. Rep., 2014.

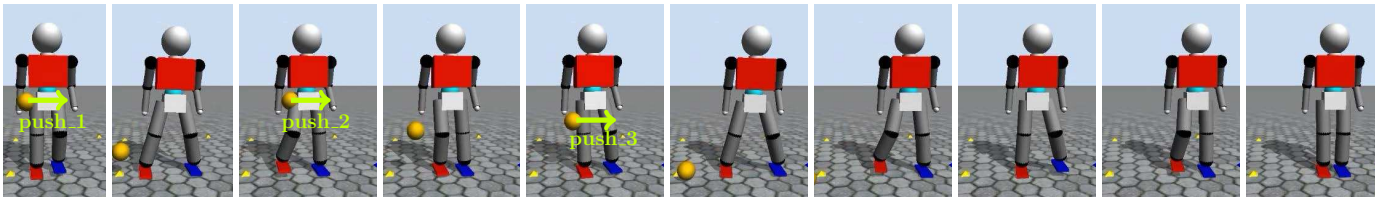


Fig. 13. Snapshots of dynamic recovery from a series of lateral pushes in ODE (interval 0.25s).

[15] R. C. Miall and D. M. Wolpert, "Forward models for physiological motor control," *Neural networks*, vol. 9, no. 8, pp. 1265–1279, 1996.
[16] B. Mehta and S. Schaal, "Forward models in visuomotor control," *Journal of Neurophysiology*, vol. 88, no. 2, pp. 942–953, 2002.
[17] D. M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," *Neural networks*, vol. 11, no. 7, pp. 1317–1329, 1998.
[18] M. Mischiati, H.-T. Lin, P. Herold, E. Imler, R. Olberg, and A. Leonardo, "Internal models direct dragonfly interception steering," *Nature*, vol. 517, no. 7534, pp. 333–338, 2015.
[19] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal, "Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics," in *International Conference on Intelligent Robots and Systems*, 2014, pp. 981–988.
[20] T. McGeer, "Passive dynamic walking," *the international journal of robotics research*, vol. 9, no. 2, pp. 62–82, 1990.
[21] J. Castano, Z. Li, C. Zhou, N. Tsagarakis, and D. Caldwell, "Dynamic and reactive walking for humanoid robots based on foot placement control," *International Journal of Humanoid Robotics*, vol. 13, p. 1550041 (44 pages), 2016.
[22] Y. You, Z. Li, N. G. Tsagarakis, and D. G. Caldwell, "Foot placement control for bipedal walking on uneven terrain: An online linear regression analysis approach," in *International Conference on Climbing and Walking Robots and Support Technologies for Mobile Machines*, 2015, p. 478.
[23] C. Zhou, Z. Li, X. Wang, N. Tsagarakis, and D. Caldwell, "Stabilization of Bipedal Walking Based on Compliance Control," *Autonomous Robots*, vol. 40, pp. 1041–1057, 2016.
[24] N. G. Tsagarakis, S. Morfey, G. M. Cerda, Z. Li, and D. G. Caldwell, "Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control," in *International Conference on Robotics and Automation*, 2013, pp. 673–678.