2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Robust Gait Synthesis Combining Constrained Optimization and Imitation Learning

Jiatao Ding[1,2], Xiaohui Xiao[1], Nikos Tsagarakis[2], Yanlong Huang[3]

Abstract— **Despite plenty of motion planning strategies have been proposed for bipedal locomotion, enhancing the walking robustness in real-world environments is still an open question. This paper focuses on robust body and leg trajectories synthesis through integrating constrained optimization with imitation learning. Specifically, we first propose a Quadratically Constrained Quadratic Programming (QCQP) algorithm to make use of the ankle strategy and stepping strategy. Based on the Linear Inverted Pendulum (LIP) model, body motion can be determined by the modulated Center of Pressure (CoP) position and step parameters (including step location and step duration). After that, we exploit an imitation learning approach Kernelized Movement Primitives (KMP) to plan robot leg motions, which allows for adapting the learned motion patterns to new situations (e.g., passing through various desired points) in a straightforward manner. Several LIP simulations and whole-body dynamic simulations demonstrate that higher walking robustness can be achieved using our framework.**

I. Introduction

Humanoid robots are expected to accomplish various tasks in unstructured environments, which consequently brings significant challenges to walking motion planning as well as stabilization. In order to improve walking robustness, body and leg motions need to be synthesized online in order to respond timely to external disturbances.

In a typical framework of humanoid motion planning, the step parameters are usually first determined, and then robot body and leg movements complying with certain stability criteria are generated. Tracking the pre-defined step parameters, the utilization of ankle strategy (i.e., manipulating the Center of Pressure (CoP) movement inside the support polygon) contributes to gain robust walking patterns [1]–[3]. However, in this way, the CoP movement is limited by the finite-sized support feet, which in turn weakens the walking robustness.

Note that step parameters modulation in real-time, as an effective way for recovering from severe disturbances, has been widely studied in the past few decades. In particular, various analytic solutions have been proposed for step parameters adjustment [4]–[8]. In order to guarantee the feasibility, the physical constraints (e.g., due to the sudden changes of step location and duration) arising in the walking process should be taken into account carefully. To do so, the constrained optimization technique was adopted for modulating step parameters online. For example, Model Predictive Control (MPC) strategies capable of using future information have been applied for step location adjustment [9]–[12] and step duration modulation [13]. Nonlinear Optimization (NLP) techniques were employed to compute optimal step parameters, e.g., [14]–[16]. However, the NLP is often time-consuming because of non-linearity caused by step duration modulation. To reduce the computational cost, further works on simplified NLPs were proposed in [17]–[19], where duration-related variables were introduced to replace the exponential term of time.

Recently, to further improve the robustness, some works have been done to integrate the ankle strategy with stepping strategy. By taking [17] as a basic optimizer, the works in [20] and [21] manipulated the Zero Moment Point (ZMP) movement to fully use the allowable support region. In [18] and [22], the hierarchical structure was also proposed to realize CoP movement and step parameters adjustment.

Once CoP motion and step parameters are determined, the robot body movement can be synthesized by using the relationship revealed in dynamic models, such as the Linear Inverted Pendulum (LIP) model [23]. After that, another key-point for gait generation is to plan the feasible leg trajectories. Generally, this problem can be solved by interpolation schemes, e.g., the high order polynomial [2] and Bezier curves [24], [25]. However, this kind of interpolation is prone to over-fitting or under-fitting when via-/end- point or time duration changes dramatically (e.g., due to external disturbances) within one step.

As a promising solution to generate proper robot leg movements, motion primitives have been used in recent works [26]–[28]. However, the optimal adjustments of robot motions or step parameters that compensate for dynamic disturbances should be determined beforehand. To cope with these prerequisites, Reinforcement Learning (RL) was introduced into the Dynamic Movement Primitive (DMP) algorithm [29]. Following this idea, the task-space DMP [30] and the joint-space DMP [31] were employed for push recovery. However, it is often time-consuming to find the optimal policy using RL approaches. Specifically, DMP is limited to trajectory adaption towards a target point, where via-point modulation is not allowed. Moreover, DMP can only guarantee the converged velocity as zero while the velocity over the execution process is not controllable.

In this paper, we attempt to provide a novel framework for robust gait synthesis. Unlike those works that used the RL technique to train walking patterns, our framework con-

[1]School of Power and Mechanical Engineering, Wuhan University, Wuhan, Hubei Province, P. R. China 430072. jtding@whu.edu.cn; xhxiao@whu.edu.cn.

[2]Humanoid and Human Centered Mechatronics Research Line, Istituto Italiano di Tecnologia, via Morego, 30, Genova, Italy 16163. name.surname@iit.it

[3]School of Computing, University of Leeds, Leeds LS29JT, UK. y.l.huang@leeds.ac.uk

sists of the constrained optimization and imitation learning technique. Specifically, we propose a Quadratically Constrained Quadratic Programming (QCQP) algorithm for step parameters adjustment and CoP manipulation. In this way, the proposed strategy can make use of ankle strategy and stepping strategy for disturbance compensation. Meanwhile, the optimal body movements can also be synthesized based on the LIP model. Then, using the modulated step parameters (including step location and step duration), the swing leg trajectories are generated by using the state-of-the-art imitation learning algorithm, Kernelized Movement Primitives (KMP) [32], [33]. Compared with the aforementioned studies, e.g. [17]–[19], the main advantages of our framework include:

(*i*) the optimal step parameters and CoP positions under dynamic disturbances are determined by QCQP, while taking into account feasibility constraints. Differing from the works in [18] and [22], the ankle strategy and stepping strategy can be activated flexibly;

(*ii*) unlike DMP [34], KMP can adapt leg trajectories towards arbitrary online-generated desired points (e.g., via-points and end-points) as well as step duration in real-time;

(*iii*) through introducing duration-related substitution variables, the constrained optimization problem can be solved in a more efficient way. Besides, KMP provides an analytic solution for leg trajectories generation. Thus, the overall computational cost is suppressed to a very low level.

The rest of this paper is organized as follows. We first introduce the framework of our work in Section II. Subsequently, we propose the QCQP strategy for CoP position and step parameters modulation in Section III. In Section IV, the principle of KMP is presented. After that, LIP simulations and dynamic simulations are provided in Section V. Finally, conclusions are drawn in Section VI.

II. Overview of the Proposed Framework

Taking the reference step parameters, including reference step length s_x^{ref}, step width s_y^{ref}, step duration T^{ref} and others, as inputs, the reference robot status can be determined by using the simplify dynamic models, such as LIP model. However, due to the modelling errors and external disturbances, real robot states usually deviate from the reference ones. Therefore, in each control loop, QCQP is employed to simultaneously modulate step parameters (i.e., step length s_x, step width s_y and step duration T) and the CoP movement ($[p_x, p_y]^T$), aiming at tracking desired step parameters and Center of Mass (CoM) state. At the same time, the body movement (characterized by CoM trajectories $[c_x, c_y, c_z]^T$ when using LIP model) will also be determined.

Once the step parameters are determined, swing leg trajectories ($[x_{sw}, y_{sw}, z_{sw}]^T$) are generated via KMP, where desired points (e.g., via-points) are introduced to avoid self-collision or collisions with the ground. It is worth mentioning that the adapted leg trajectories can be obtained directly by using KMP, without the need of collecting various motion demonstrations under different walking modes.

Finally, using lower-level balance controllers, the robust walking can be ultimately accomplished. Since the tracking control problem lies outside the scope of this paper, readers are advised to refer to the admittance controller in [35]. An overview of the framework is illustrated in Fig. 1.

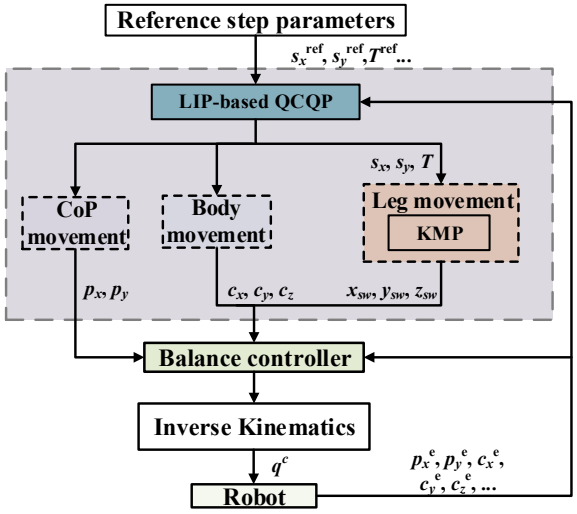

Fig. 1: Overview of the proposed framework. Based on the sensory information, QCQP is employed to manipulate the CoP position, adjust step parameters and modulate the body movement. Subsequently, KMP adapts the leg motions (i.e., swing leg trajectories) using updated step parameters.

III. Robust Body Motion Generation

As explained in Section II, robot body motions should be re-generated in real-time in order to obtain robust walking patterns. In particular, when unexpected disturbances, such as severe external pushes, are imposed to robots, step parameters and CoP positions need to be adjusted to maintain balance. In this section, we propose a QCQP algorithm to integrate ankle strategy with stepping strategy.

A. LIP motion

By employing the LIP model, robot body motions are characterized by the CoM movements. As shown in Fig. 2, considering the CoP motion, the CoM motion is determined by

$$\begin{aligned} {}_\iota p_\gamma &= {}_\iota c_\gamma - \frac{1}{\omega^2}\,{}_\iota \ddot{c}_\gamma, \gamma \in \{x, y\}, \\ \omega &= \sqrt{g/Z_{\text{c}}}, \end{aligned} \tag{1}$$

where ${}_\iota c_\gamma$ denotes the horizontal CoM position (relative to the current support center, i.e., local coordinate), ${}_\iota \ddot{c}_\gamma$ denotes the CoM acceleration, ${}_\iota p_\gamma$ denotes the relative CoP position, ω is the natural frequency, g is the gravitational acceleration and Z_{c} is the constant pendulum height,

Assuming the fixed CoP position during the whole step, the CoM state can be determined by

$$\begin{aligned} {}_\iota c_{\gamma(t_e)} &= {}_\iota c_{\gamma(0)} \cosh(\omega t_e) + \frac{{}_\iota \dot{c}_{\gamma(0)}}{\omega} \sinh(\omega t_e) + {}_\iota p_\gamma, \\ {}_\iota \dot{c}_{\gamma(t_e)} &= {}_\iota c_{\gamma(0)} \omega \sinh(\omega t_e) + {}_\iota \dot{c}_{\gamma(0)} \cosh(\omega t_e), \end{aligned} \tag{2}$$

where t_{e} is the elapsed time, ${}_\iota c_{\gamma(t_e)}$ and ${}_\iota \dot{c}_{\gamma(t_e)}$ separately denote the current CoM position and velocity, ${}_\iota c_{\gamma(0)}$ and

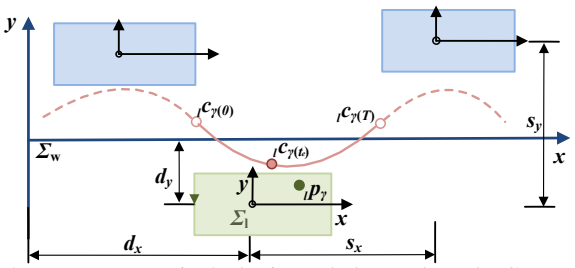

Fig. 2: LIP movement in the horizontal plane, where the CoM movement is determined by the step parameters and CoP movements. The red solid curve plots the CoM trajectory of current step and the red dash curves respectively depict the CoM trajectories of the previous and next steps. $[d_x, d_y]^T$ denotes the global step location, s_x and s_y are the step length and step width, Σ_w and Σ_l represent the origins of the global and local coordinates, respectively.

$_lc_{\gamma(0)}$ denote the initial CoM position and velocity, sinh and cosh denote the hyperbolic sine and cosine function.

Given the current state (i.e., CoM position, CoM velocity and CoP position) at time t_e, the final state is predicted as

$$\begin{aligned} {}_lc_{\gamma(T)} &= ({}_lc_{\gamma(t_e)} - {}_lp_\gamma)\cosh(\omega(\delta T)) + \frac{{}_l\dot{c}_{\gamma(t_e)}}{\omega}\sinh(\omega(\delta T)) + {}_lp_\gamma, \\ {}_l\dot{c}_{\gamma(T)} &= ({}_lc_{\gamma(t_e)} - {}_lp_\gamma)\omega\sinh(\omega(\delta T)) + {}_l\dot{c}_{\gamma(t_e)}\cosh(\omega(\delta T)), \end{aligned} \tag{3}$$

where $\delta T = T - t_e$ is the remaining time of current step (T is the time duration), ${}_lc_{\gamma(T)}$ and ${}_l\dot{c}_{\gamma(T)}$ separately denote the final CoM position and velocity of current cycle.

As done in [18] and [19], duration-related variables are defined as

$$t_{\text{ch}} = \cosh(\omega(\delta T)), \quad t_{\text{sh}} = \sinh(\omega(\delta T)), \tag{4}$$

which can be used to rewrite (3) in a compact form

$$\begin{aligned} {}_lc_{\gamma(T)} &= ({}_lc_{\gamma(t_e)} - {}_lp_\gamma)t_{\text{ch}} + \frac{{}_l\dot{c}_{\gamma(t_e)}}{\omega}t_{\text{sh}} + {}_lp_\gamma, \\ {}_l\dot{c}_{\gamma(T)} &= ({}_lc_{\gamma(t_e)} - {}_lp_\gamma)\omega t_{\text{sh}} + {}_l\dot{c}_{\gamma(t_e)}t_{\text{ch}}. \end{aligned} \tag{5}$$

Obviously, as time goes by, the t_e increases from 0 to T. Thus, the δT decreases from T to 0. In this case, the monotonicities of the hyperbolic sine and cosine functions are guaranteed. As a result, we can exploit the duration-related variables to simplify LIP motion equations.

Observing (5), we can find that the CoM motion is fully determined by two conditions, if there is no CoP motions during the whole step (i.e., ${}_lp_\gamma$ is constant). In this paper, to track the reference step parameters and guarantee the continuity of the CoM trajectory, we choose the current sensing CoM position and the final reference position of CoM as boundary conditions, i.e.,

$${}_lc_{\gamma(t_e)} = {}_lc^{\text{e}}_{\gamma(t_e)}, \quad {}_lc_{\gamma(T)} = s_\gamma/2, \tag{6}$$

where ${}_lc^{\text{e}}_{\gamma(t_e)}$ is the estimated CoM position, s_γ is the current step length or step width. Intuitively, ${}^is_\gamma/2$ is set to be the reference final CoM position of current step.

B. QCQP formulation

In [19], by employing the duration-related variables, a constrained optimization problem was formulated to modulate step location and step duration, where the NLP strategy, however, ignored the CoP movement and thus failed to utilize the support region caused by foot size. In contrast, by taking into account the feasibility constraints, we exploit QCQP technique to accomplish robust walking by integrating ankle strategy with stepping strategy.

1) Objective Function: In order to accomplish desired walking tasks, we propose to track the desired step parameters, desired final CoM states and desired CoP movement. Particularly, to simplify this problem, we only optimize the system state of current step[1], yielding a minimization problem

$$f(\boldsymbol{\mathcal{X}}) = \sum_{\mathbf{X}} \frac{\sigma_{\mathbf{X}}}{2} \parallel \mathbf{X} - \mathbf{X}^{\text{ref}} \parallel^2, \tag{7}$$

where $\boldsymbol{\mathcal{X}} = [s_x, s_y, t_{\text{ch}}, t_{\text{sh}}, {}_lp_x, {}_lp_y]^T$ consists of step length (s_x), step width (s_y), step-duration related variables (t_{ch}, t_{sh}) and CoP position (p_x, p_y) of current step. $\mathbf{X} = [\boldsymbol{\mathcal{X}}^T, {}_lc_{x(T)}, {}_lc_{y(T)}, {}_l\dot{c}_{x(T)}, {}_l\dot{c}_{y(T)}]$ denotes the variables contributing to the cost terms. $\sigma_{\mathbf{X}}$ denotes the weight coefficient. $\mathbf{X}^{\text{ref}}$ denotes the reference value. For example, the ${}_lc^{\text{ref}}_{x(T)}$, ${}_lc^{\text{ref}}_{y(T)}$, $t^{\text{ref}}_{\text{ch}}$, $t^{\text{ref}}_{\text{ch}}$, ${}_lp^{\text{ref}}_x$ and ${}_lp^{\text{ref}}_y$ are given as

$$\begin{cases} {}_lc^{\text{ref}}_{x(T)} = s^{\text{ref}}_x/2, & {}_lc^{\text{ref}}_{y(T)} = s^{\text{ref}}_y/2, \\ t^{\text{ref}}_{\text{ch}} = \cosh(\omega(\delta T^{\text{ref}})), & t^{\text{ref}}_{\text{sh}} = \sinh(\omega(\delta T^{\text{ref}})), \\ {}_lp^{\text{ref}}_x = 0, & {}_lp^{\text{ref}}_y = 0, \end{cases} \tag{8}$$

where $\delta T^{\text{ref}} = T^{\text{ref}} - t_{\text{e}}$ (T^{ref} denotes the reference time duration of current step), s^{ref}_x and s^{ref}_y are the reference step length and step width of current step.

Since the robot is expected to track the absolute step location, the reference step length and step width for the current (i^{th}) step are determined as

$${}^is^{\text{ref}}_x = {}^{i+1}d^{\text{ref}}_x - {}^id_x, \quad {}^is^{\text{ref}}_y = {}^{i+1}d^{\text{ref}}_y - {}^id_y, \tag{9}$$

where $[{}^{i+1}d^{\text{ref}}_x, {}^{i+1}d^{\text{ref}}_y]^T$ is the desired step location of next step cycle, which is calculated by adding the pre-defined step parameters recursively. $[{}^id_x, {}^id_y]^T$ is the optimal step location of current step, which is calculated by adding the real-time generated step parameters recursively.

Note that all the cost terms in (7) can be expressed as quadratic forms of the optimal variables, excepting for the tracking error terms defined on the final CoM position and velocity, where the multiplication operation between ${}_lp_\gamma$ and $t_{\text{ch}}(t_{\text{sh}})$ appear, as expressed in (5). To simplify this problem, we replace the ${}_lp_\gamma$, that was involved in the multiplication operation (in (5)), by the estimated CoP position. As a result, the objective function (7) can be formed as a quadratic function of the optimal variables, i.e.,

$$f(\boldsymbol{\mathcal{X}}) = \frac{1}{2}\boldsymbol{\mathcal{X}}^T\mathbf{G}\boldsymbol{\mathcal{X}} + \mathbf{g}^T\boldsymbol{\mathcal{X}}, \tag{10}$$

[1]It has been proved that, if the CoM state is *viable*, adjusting one step is enough to reject severe dynamic disturbances [17]. In [19], we also found that, in some cases, one-step-prediction could gain almost the same push recovery capability as the two-steps-prediction. Thus, considering the CoP movement, we merely adjust the step parameters of current step.

where $\mathbf{G} \in \Re^{6\times 6}$ and $\mathbf{g} \in \Re^{6}$ are defined as

$$\begin{aligned}\mathbf{G} &= \sigma_{s_x}\mathbf{S}_1^T\mathbf{S}_1+\sigma_{s_y}\mathbf{S}_2^T\mathbf{S}_2+\sigma_{t_{\text{ch}}}\mathbf{S}_3^T\mathbf{S}_3+\sigma_{t_{\text{sh}}}\mathbf{S}_4^T\mathbf{S}_4+\sigma_{{}_lp_x}\mathbf{S}_5^T\mathbf{S}_5\\ &+\sigma_{{}_lp_y}\mathbf{S}_6^T\mathbf{S}_6+\sigma_{{}_lc_{x(T)}}\mathbf{H}_{{}_lc_{x(T)}}^T\mathbf{H}_{{}_lc_{x(T)}}+\sigma_{{}_lc_{y(T)}}\mathbf{H}_{{}_lc_{y(T)}}^T\mathbf{H}_{{}_lc_{y(T)}}\\ &+\sigma_{{}_l\dot{c}_{x(T)}}\mathbf{H}_{{}_l\dot{c}_{x(T)}}^T\mathbf{H}_{{}_l\dot{c}_{x(T)}}+\sigma_{{}_l\dot{c}_{y(T)}}\mathbf{H}_{{}_l\dot{c}_{y(T)}}^T\mathbf{H}_{{}_l\dot{c}_{y(T)}},\\ \mathbf{g} &= -(\sigma_{s_x}s_x^{\text{ref}}\mathbf{S}_1+\sigma_{s_y}s_y^{\text{ref}}\mathbf{S}_2+\sigma_{t_{\text{ch}}}t_{\text{ch}}^{\text{ref}}\mathbf{S}_3+\sigma_{t_{\text{sh}}}t_{\text{sh}}^{\text{ref}}\mathbf{S}_4\\ &+\sigma_{{}_lc_{x(T)}}{}_lc_{x(T)}^{\text{ref}}\mathbf{H}_{{}_lc_{x(T)}}+\sigma_{{}_lc_{y(T)}}{}_lc_{y(T)}^{\text{ref}}\mathbf{H}_{{}_lc_{y(T)}}\\ &+\sigma_{{}_l\dot{c}_{x(T)}}{}_l\dot{c}_{x(T)}^{\text{ref}}\mathbf{H}_{{}_l\dot{c}_{x(T)}}+\sigma_{{}_l\dot{c}_{y(T)}}{}_l\dot{c}_{y(T)}^{\text{ref}}\mathbf{H}_{{}_l\dot{c}_{y(T)}})^T.\end{aligned} \tag{11}$$

Here, $\mathbf{S}_j \in \Re^{1\times 6}(j \in 1, 2, \cdots, 6)$ is the variable selective matrix[2], $\mathbf{H}_{{}_lc_{x(T)}}$, $\mathbf{H}_{{}_lc_{y(T)}}$, $\mathbf{H}_{{}_l\dot{c}_{x(T)}}$, and $\mathbf{H}_{{}_l\dot{c}_{y(T)}}$ are the matrices about the reference CoM state, which are given by

$$\begin{aligned}\mathbf{H}_{{}_lc_{x(T)}} &= ({}_lc_{x(t_e)}^{\text{e}} - {}_lp_{x(t_e)}^{\text{e}})\mathbf{S}_3 + \frac{1}{\omega}{}_l\dot{c}_{x(t_e)}^{\text{e}}\mathbf{S}_4 + \mathbf{S}_5,\\ \mathbf{H}_{{}_lc_{y(T)}} &= ({}_lc_{y(t_e)}^{\text{e}} - {}_lp_{y(t_e)}^{\text{e}})\mathbf{S}_3 + \frac{1}{\omega}{}_l\dot{c}_{y(t_e)}^{\text{e}}\mathbf{S}_4 + \mathbf{S}_6,\\ \mathbf{H}_{{}_l\dot{c}_{x(T)}} &= ({}_lc_{x(t_e)}^{\text{e}} - {}_lp_{x(t_e)}^{\text{e}})\omega\mathbf{S}_4 + {}_l\dot{c}_{x(t_e)}^{\text{e}}\mathbf{S}_3,\\ \mathbf{H}_{{}_l\dot{c}_{y(T)}} &= ({}_lc_{y(t_e)}^{\text{e}} - {}_lp_{y(t_e)}^{\text{e}})\omega\mathbf{S}_4 + {}_l\dot{c}_{y(t_e)}^{\text{e}}\mathbf{S}_3,\end{aligned} \tag{12}$$

with $[{}_lc_{x(t_e)}^{\text{e}}, {}_lc_{y(t_e)}^{\text{e}}]^T$, $[{}_l\dot{c}_{x(t_e)}^{\text{e}}, {}_l\dot{c}_{y(t_e)}^{\text{e}}]^T$ and $[{}_lp_{x(t_e)}^{\text{e}}, {}_lp_{x(t_e)}^{\text{e}}]^T$ being the estimated CoM position, CoM velocity and CoP position, respectively.

2) Feasibility Constraints: To guarantee the feasibility, constraints arising from the physical limitations of actuation capability, mechanical structure and environmental scenarios should be taken into account when planning walking patterns. In this paper, the feasibility constraints for the NLP formulation defined in [19] are used. Specifically, these constraints consist of the limitations on step duration, step location, swing leg velocities, as well as those on the high-order derivatives of CoM state (e.g., CoM acceleration and jerk). Besides, the CoP should stay within the support polygon. To be brief, only the constraints on step duration variation and CoP movements are introduced here.

Constraints of step duration variation: The step frequency, determined by the step duration, is limited by the actuation capability. First, given the lower boundary $T^{\min}$ and upper boundary $T^{\max}$ of the step duration, we can derive the constraints on variables t_{ch} and t_{sh} by utilizing the monotonicity of hyperbolic functions. For example, the max-min constraint of t_{ch} is given as

$$\cosh(\omega(\max(T^{\min}-t_{\text{e}},0))) \le t_{\text{ch}} \le \cosh(\omega(T^{\max}-t_{\text{e}})), \tag{13}$$

where $\max()$ function returns the maximal value among input variables.

Additionally, considering the property of hyperbolic functions, the following equality constraint should be satisfied

$$t_{\text{ch}}^2 - t_{\text{sh}}^2 = 1. \tag{14}$$

Constraints of CoP movement: The CoP movement should be restricted within the support region so as to guarantee the walking stability. Herein, we consider the max-min constraint on CoP movement, expressed as

$${}_lp_\gamma^{\min} \le {}_lp_\gamma \le {}_lp_\gamma^{\max}, \tag{15}$$

[2]All the elements in $\mathbf{S}_j$ are zeros except that the j^{th} element is 1.

where ${}_lp_\gamma^{\min}$ and ${}_lp_\gamma^{\max}$ respectively denote the lower boundary and upper boundary of CoP position.

Thus, the feasibility constraints can be expressed in the form of quadratic inequalities and accordingly the NLP problem can be formulated as a QCQP problem, which can be solved by Sequential Quadratic Programming (SQP). Please refer to [12] for more details.

IV. Robust Leg Motion Generation

Assuming that we have obtained the optimal step parameters by deploying the QCQP optimizer, we now focus on the generation of robust leg trajectories. As explained before, several key features should be taken into account when planning leg trajectories: (*i*) sudden changes of step parameters (including step locations and step duration) should be incorporated and (*ii*) feasible constraints should be satisfied, including collision avoidance and landing impact reduction.

In order to generate feasible leg trajectories, we propose to generate swing leg trajectories by resorting to the imitation learning strategy, where robot motions can be generated by learning from a few demonstrations which are collected from human or robot walking experiments. Since the supporting leg keeps static when the swing leg is moving during one step, we consider only the swing leg trajectory relative to support leg.

Now, we explain the principle of KMP which we apply in our framework. Formally, assuming that we have access to a set of demonstrations (i.e., training data) $\mathbf{D} = \{\{t_{n,h}, \boldsymbol{\xi}_{n,h}\}_{n=1}^N\}_{h=1}^H$, with $\boldsymbol{\xi}_{n,h} \in \Re^3$ denoting the Cartesian trajectory point at the n^{th} time step from the h^{th} demonstrations. Here, N and H separately denote the trajectory length and the number of demonstrations. Then, we can use Gaussian Mixture Model (GMM) to model the joint probability distribution $\mathcal{P}(t, \boldsymbol{\xi})$ [33], [36], yielding

$$\begin{bmatrix} t \\ \boldsymbol{\xi} \end{bmatrix} \sim \sum_{c=1}^{C} \pi_c \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c), \tag{16}$$

where π_c, $\boldsymbol{\mu}_c = \begin{bmatrix} \boldsymbol{\mu}_{t,c} \\ \boldsymbol{\mu}_{\eta,c} \end{bmatrix}$ and $\boldsymbol{\Sigma}_c = \begin{bmatrix} \boldsymbol{\Sigma}_{tt,c} & \boldsymbol{\Sigma}_{t\eta,c} \\ \boldsymbol{\Sigma}_{\eta t,c} & \boldsymbol{\Sigma}_{\eta\eta,c} \end{bmatrix}$ respectively correspond to the prior probability, mean and covariance of the c-th Gaussian component in GMM. Subsequently, Gaussian Mixture Regression (GMR) [33], [36], [37] is used to retrieve a probabilistic reference trajectory $\mathbf{D}_r = \{t_n, \hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n\}_{n=1}^N$, which in fact encapsulates the distribution of demonstrations and will be used to train KMP.

Let us denote

$$\begin{aligned}\boldsymbol{\Sigma} &= blockdiag(\hat{\boldsymbol{\Sigma}}_1, \hat{\boldsymbol{\Sigma}}_2, \ldots, \hat{\boldsymbol{\Sigma}}_N),\\ \mathbf{U} &= [\hat{\boldsymbol{\mu}}_1\, \hat{\boldsymbol{\mu}}_2\, \cdots\, \hat{\boldsymbol{\mu}}_N].\end{aligned} \tag{17}$$

For an arbitrary inquiry input t^*, KMP predicts the corresponding output as

$$\boldsymbol{\xi}(t^*) = \mathbf{k}^*(\mathbf{K} + \lambda\boldsymbol{\Sigma})^{-1}\mathbf{U} \tag{18}$$

with

$$\mathbf{k}^* = \begin{bmatrix}\mathbf{k}(t^*, t_1)\ \mathbf{k}(t^*, t_2)\ \ldots\ \mathbf{k}(t^*, t_N)\end{bmatrix} \tag{19}$$

and

$$\mathbf{K} = \begin{bmatrix} \mathbf{k}(t_1,t_1) & \mathbf{k}(t_1,t_2) & \dots & \mathbf{k}(t_1,t_N) \\ \mathbf{k}(t_2,t_1) & \mathbf{k}(t_2,t_2) & \dots & \mathbf{k}(t_2,t_N) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{k}(t_N,t_1) & \mathbf{k}(t_N,t_2) & \dots & \mathbf{k}(t_N,t_N) \end{bmatrix}, \quad (20)$$

where $\mathbf{k}(t_i,t_j) = k(t_i,t_j)\mathbf{I}_3$ with $k(\cdot,\cdot)$ being a kernel function (e.g., Gaussian kernel).

It is worth mentioning that the trajectory adaptation issue can be straightforwardly solved by KMP through concatenating the original reference trajectory with the desired points. For example, given L desired points, described by $\bar{\mathbf{D}} = \{\bar{t}_l, \bar{\boldsymbol{\mu}}_l, \bar{\boldsymbol{\Sigma}}_l\}_{l=1}^{L}$, where $\bar{\boldsymbol{\mu}}_l$ and $\bar{\boldsymbol{\Sigma}}_l$ correspond to the mean and covariance of the desired point at the time $\bar{t}_l$, we can simply combine $\bar{\mathbf{D}}$ with $\mathbf{D}$, resulting to an extended reference trajectory $\tilde{\mathbf{D}} = \bar{\mathbf{D}} \cup \mathbf{D}$. After that, by exploiting $\tilde{\mathbf{D}}$ instead of $\mathbf{D}$ in (18)-(20), KMP is capable of generating adapted trajectories that pass through all desired points and meanwhile maintaining the shape of demonstrations. We suggest readers refer to [32], [33] for the details of KMP.

V. Evaluations

To verify our framework, the LIP and whole-body dynamic simulations are conducted on the prototype of a 69kg weighted humanoid robot "COMAN+", which is 1.6m in height and actuated by compliant Series Elastic Actuators (SEAs). For bipedal walking, the default step length, step width and step duration are respectively set as 0.1m, 0.206m and 0.7s. Besides, the clear height of the swing foot (l_h) is 0.07m, and constant LIP height (Z_c) is 0.89m. The sampling time for QCQP solution (dt) is 0.05s. Other relevant parameters are listed in Appendix.

For the leg motion generation, we used 5 demonstrations under subject #8 (walking mode) from the motion capture database released by *the Graphics Lab in Carnegie Mellon University* [38] to train KMP. Based on modulated step parameters, the adapted leg trajectory was required to track a via-point ($[{}^i d_x, {}^i d_y, l_h]^T$) at the middle of one step while track the end-point ($[{}^{i+1} d_x, {}^{i+1} d_y, 0]^T$) at the end of one step[3]. Besides, the desired velocity when passing through the via-point was set to be $[({}^{i+1}d_x - {}^{i-1}d_x)/T, ({}^{i+1}d_y - {}^{i-1}d_y)/T, 0]^T$ out of experience while the velocity when passing through the end-point was set to be $[0,0,0]^T$ for reducing landing impact. Note that the single support phase took up the 80% of one step.

A. LIP simulation

In this section, we test our framework through the push recovery evaluations. Without loss of the generality, the multi-directional horizontal pushes were imposed on the pelvis, lasting 0.1s. Specifically, we imposed a 300N forward force and a 225N leftward force at 2.1s, followed by a 300N backward force and a 225 N rightward force at 4.5s. The generated motions by using our framework are demonstrated in Fig. 3 - Fig. 5.

[3]That is to say, the via-point and end-point would be modulated as the step parameters change

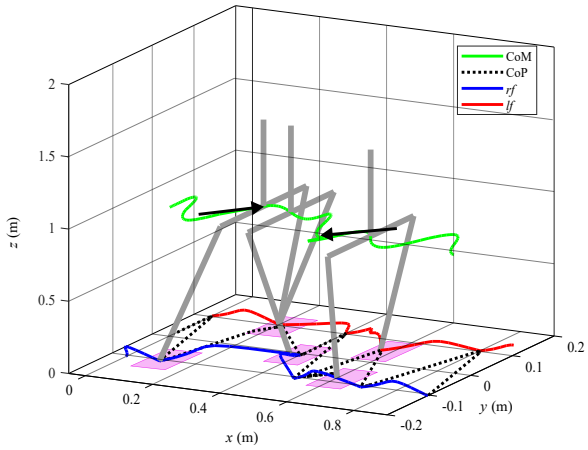

Fig. 3: The robot recovered from external pushes. Black arrows mark the directions of external pushes. Pink rectangles represent the adjusted footprints.

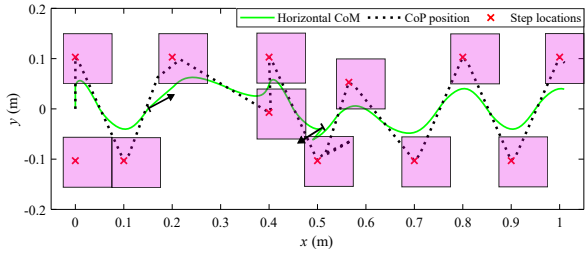

Fig. 4: Horizontal CoM trajectory, CoP trajectory and step locations for push-recovery. The ankle strategy and stepping strategy were activated while the CoP was restricted within the support region.

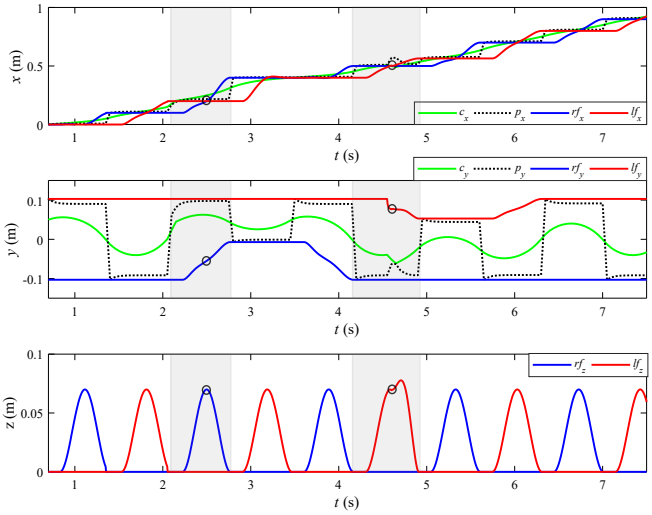

Fig. 5: Motion components for push-recovery. $[c_x, c_y]^T$ and $[p_x, p_y]^T$ separately denote the global CoM position and CoP position, $[rf_x, rf_y, rf_z]^T$ and $[lf_x, lf_y, lf_z]^T$ separately denote the right and left leg trajectories. Gray zones cover the periods where step duration suddenly changes. The black circles mark the calculated via-points at the step when the push force was imposed.

TABLE I: Modulated step parameters for push recovery

Paras \ step	1	4	5	7	8	others
$s_x(m)$	0	0.2	0	0.065	0.135	0.1
$\lvert s_y \rvert (m)$	0.206	0.11	0.11	0.156	0.156	0.206
$T(s)$	0.699	0.674	0.699	0.744	0.699	0.699

1) Body motion generation: As can be seen in Fig. 3 and Fig. 4, when the left-forward push was applied at 2.1s, the robot landed on the ground with a larger step length (variation from 0.1m to 0.2m). Since the robot was currently supported by the left leg, the smaller step width was also used (from 0.206m to 0.11m). Besides, the robot also reduced the step period so as to compensate for the disturbance. As can be seen in Table I, the step duration of the 4^{th} step reduced to be 0.674s. Apart from the stepping strategy, the CoP position was also modulated (deviated from the support center as shown in Fig. 4 and Fig. 5), which contributed to higher walking stability. After that, when the right-backward force was applied, the robot shortened the step length to 0.065m and reduced the step width to 0.156m (right leg supports the whole body at this step). At the same time, the robot extended the step duration to 0.744s in order to absorb the perturbation. In addition, as can be seen in Fig.4, the CoP trajectory was also modulated.

Since the objective function aims to track the global step location, the 5^{th} step length reduced to be 0m while the step width reduced to be 0.11m. As a result, the robot returned to the reference status after two-steps modulation. The same phenomenon can also be observed at the 8^{th} step.

2) Leg trajectory generation: As shown in Fig. 5, KMP was capable of generating feasible leg trajectories according to the modulated step parameters. Specifically, KMP can reserve the shape of demonstrated motions (especially the shape of height trajectory) whenever a shorter step duration (the 4^{th} step) or a longer one (the 7^{th} step) is required, where the under-fitness or over-fitness is avoided.

In addition, when subjecting to variant desired points (via-points and end-points), the adaptive leg motions could still be generated, which can pass through the desired via-points and end-points precisely, as can be seen in Fig. 5. Detailed analysis revealed that the adapted trajectories also satisfied the velocity constraints at the via-points and end-point points. That is to say, the KMP can adapt leg trajectories towards online-generated desired points via desired velocities, which can not be accomplished by DMP to our knowledge.

B. Whole-body Dynamic Simulation

The whole-body dynamic simulations were carried out by using the open-source simulator *Gazebo*. The control system was based on XbotCore [39]. Written in $c++$ language, each optimization loop can be computed within 0.5ms on a 3.0 GHz CPU. The balance controller is run in 500 Hz.

1) Non-periodic walking: Modelling errors are usually inevitable when using the LIP model, such as those arising from the distributed mass of the swing leg and the intrinsic compliance of the robot structure. To achieve higher robustness in real-world environments, the robot should be able to accomplish different walking tasks while taking into account modelling errors. In this section, the robot is controlled to accomplish non-periodic walking with time-varying step lengths (as listed in Table II), consisting of forward and backward motions.

TABLE II: Step length setup for non-periodic walking

Paras \ step	1	2	3-5	6	7-11	12
$s_x\,(\mathrm{m})$	0	0.05	0.1	0	-0.05	0

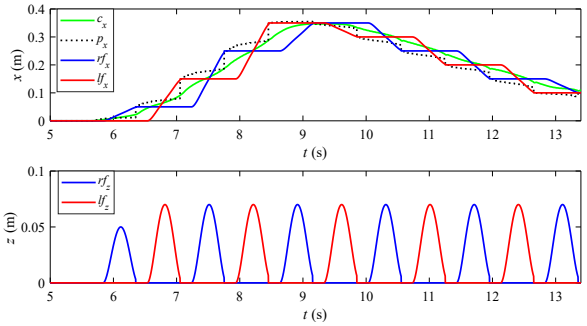

Fig. 6: Generated trajectories for non-periodic walking. The clear height of swing leg for startup was adjusted to be 0.05m due to the modelling errors.

As can be seen in Fig. 6, the smooth CoM trajectory was obtained by the proposed QCQP method, where the CoP movement was adjusted. Notably, the leg movements for backward walking could also be obtained using KMP, even only these demonstrations corresponding to the forward walking were used to train KMP. The snapshots of evaluations are depicted in Fig. 7. Please see also the supplementary video for the details of these evaluations.

2) Recovery from external pushes: Now, we evaluate the push recovery capability of our framework, where the robot is stepping in place while various horizontal forces were applied on the pelvis, with each force lasting 0.1s.

Push recovery from lateral pushes: We first applied a 150N rightward force at 9.9s and subsequently a 150N leftward force at 12.5s. By employing the proposed framework, the ankle and stepping strategies were combined to maintain stability. The robot motions are demonstrated in Fig. 8 and the modulated step parameters are listed in Table III.

To compensate for the status errors caused by external pushes, the CoP position was modified, as shown in Fig. 8. For example, after the leftward push, the generated p_y deviated from the 12^{th} and 13^{th} step locations. In addition, the optimal step parameters were also modulated in real-time, as can be seen in Table III. By observing Fig. 8, we can find that the robot reduced the step duration dramatically (the step duration of the 8^{th} was reduced to 0.5s) when the rightward push was detected. Meanwhile, the step width was also reduced. Afterwards, when the leftward push was detected, the robot accelerated the step frequency while taking a smaller step width.

Differing from the LIP simulation, when the external push disappeared, the modified step widths can not guarantee that global lateral step locations were tracked accurately in each step. Nevertheless, since the step width was re-computed in each loop, the overall lateral displacement could be ignored, as depicted in Fig. 8. Note that the step parameters of the 4^{th} and 5^{th} steps were also adjusted, which compensated for the modelling errors, especially the intrinsic compliance caused

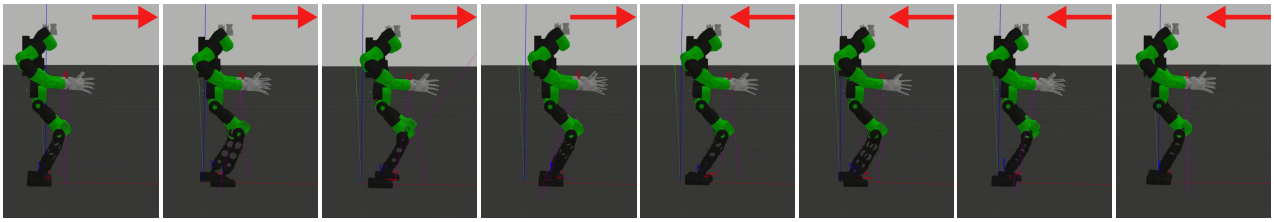

Fig. 7: Snapshots of the non-periodic walking. Red arrows depict the walking directions.

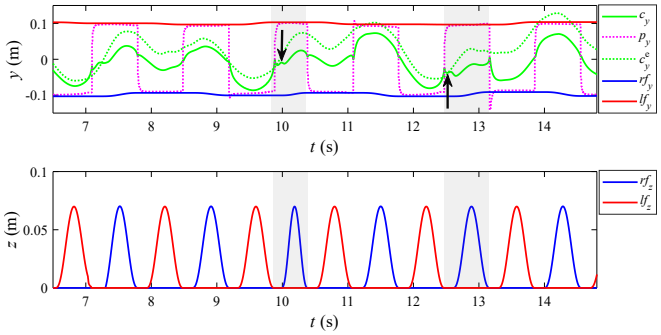

Fig. 8: Robust gait for rightward and leftward push recovery. c_y^e is the estimated forward CoM trajectory, black arrows indicate time moments when the external forces were applied.

TABLE III: Modulated step parameters for push recovery (the parameters of other steps are almost the same with reference values)

Paras \ step	4	5	8	9	12	13
$\lvert s_y \rvert$(m)	0.194	0.192	0.197	0.191	0.188	0.195
T(s)	0.692	0.695	0.5	0.696	0.687	0.697

by the compliant actuators and the long leg structure.

We also tested our framework by imposing the forward and backward pushing forces, please watch the complementary video for these evaluations.

Maximal tolerant forces: Another advantage of this work is that different balance strategies can be flexibly activated by simply setting different weight ratios for each cost term in (7). In this section, by manually tuning these parameters, we test the maximal tolerant push from which the robot can recover. The evaluations are presented in Fig 9, showing that the push rejection capability has been enhanced dramatically as a consequence of the integration of stepping strategy.

VI. Conclusions

We have proposed a novel framework, which consists of the constrained optimization and imitation learning. Using

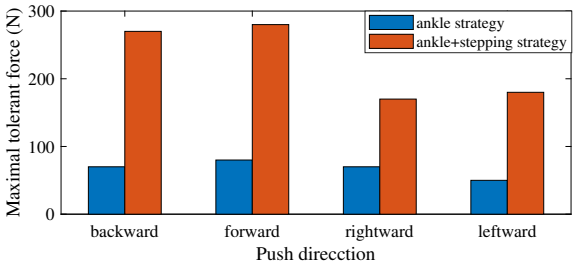

Fig. 9: Maximal tolerant forces under different strategies.

TABLE IV: Weight coefficients for QCQP solution

$\sigma_{{}_l p_x}$	$3.5\times10^7/5\times10^6$	$\sigma_{{}_l p_y}$	$4\times10^7/5\times10^6$
σ_{s_x}	$1\times10^8/3\times10^8$	σ_{s_y}	$2\times10^8/1\times10^9$
$\sigma_{t_{ch}}$	$3\times10^8/2.5\times10^7$	$\sigma_{t_{sh}}$	$3\times10^8/2.5\times10^7$
$\sigma_{{}_l c_{x(T)}}$	$5\times10^7/1\times10^6$	$\sigma_{{}_l c_{y(T)}}$	$4\times10^7/8\times10^6$
$\sigma_{{}_l \dot{c}_{x(T)}}$	$2\times10^5/8\times10^6$	$\sigma_{{}_l \dot{c}_{y(T)}}$	$5\times10^6/8\times10^5$

TABLE V: Boundary conditions for feasibility constraints

Step location constraints		Step duration constraints	
$s_x^{\min}$/m	-0.05	$T^{\min}$/s	0.5
$s_x^{\max}$/m	0.2	$T^{\max}$/s	1.2
$s_y^{\min}$/m	0.11	CoP movement constraints	
$s_y^{\max}$/m	0.26	${}_l p_x^{\min}$/m	-0.03
$\dot{s}_x^{\min}$/m·s^{-1}	-0.75	${}_l p_x^{\max}$/m	0.07
$\dot{s}_x^{\max}$/m·s^{-1}	1.5	${}_l p_y^{\min}$/m	-0.04
$\dot{s}_y^{\min}(\dot{s}_y^{\max})$/m·s^{-1}	-1	${}_l p_y^{\max}$/m	0.05

the QCQP strategy, the CoM trajectories and step parameters are modulated in real-time so as to respond timely to external disturbances, where the ankle strategy and stepping strategy are integrated. Then, the KMP algorithm is deployed to generate adaptive swing leg trajectories through learning from human walking data. Simulations on a compliant humanoid robot demonstrate the robustness of the proposed framework.

In the future, the foot rotation can be integrated into our framework. Besides, based on the presented work, the body movements and leg movements would be learned simultaneously while considering the physical constraints, with the goal of realizing human-like walking.

Acknowledgement

This work is supported by National Natural Science Foundation of China (Grant No. 51675385 and 51175383) and European Union's Horizon 2020 robotics program CogIMon (ICT-23-2014, 644727).

Appendix

The basic parameters for QCQP solution are listed in Table IV and Table V. For each weight coefficient, the right one is for LIP simulations while the left one is for the whole-body simulations. The boundary conditions for feasibility constraints are used for both LIP simulations and whole-body dynamic simulations. Several boundary conditions which are ignored in Table V can be found in [19].

REFERENCES

[1] 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 *IEEE International Conference on Robotics and Automation*, vol. 2, 2003, pp. 1620–1626.

[2] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method for real-time gait planning for humanoid robots," *International Journal of Humanoid Robotics*, vol. 3, no. 01, pp. 1–19, 2006.

[3] 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*, 2006, pp. 137–142.

[4] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa, "Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 3989–3994.

[5] J. A. 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, no. 02, pp. 1 550 041–1–1 550 041–44, 2016.

[6] J. Ding, Y. Wang, M. Yang, and X. Xiao, "Walking stabilization control for humanoid robots on unknown slope based on walking sequences adjustment," *Journal of Intelligent & Robotic Systems*, vol. 90, no. 3-4, pp. 323–338, 2018.

[7] R. J. Griffin, G. Wiedebach, S. Bertrand, A. Leonessa, and J. Pratt, "Walking stabilization using step timing and location adjustment on the humanoid robot, atlas," *arXiv preprint arXiv:1703.00477*, 2017.

[8] D. Kim, S. J. Jorgensen, J. Lee, J. Ahn, J. Luo, and L. Sentis, "Dynamic locomotion for passive-ankle biped robots and humanoids using whole-body locomotion control," *The International Journal of Robotics Research*, 2020. DOI: 10.1177/0278364920918014.

[9] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl, "Online walking gait generation with adaptive foot positioning through linear model predictive control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1121–1126.

[10] R. J. Griffin and A. Leonessa, "Model predictive control for dynamic footstep adjustment using the divergent component of motion," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1763–1768.

[11] A.-C. Hildebrandt, R. Wittmann, F. Sygulla, D. Wahrmann, D. Rixen, and T. Buschmann, "Versatile and robust bipedal walking in unknown environments: real-time collision avoidance and disturbance rejection," *Autonomous Robots*, pp. 1–20, 2019.

[12] J. Ding, C. Zhou, S. Xin, X. Xiao, and N. Tsagarakis, "Nonlinear model predictive control for robust bipedal locomotion exploring com height and angular momentum changes," *arXiv preprint arXiv:1902.06770*, 2019.

[13] Z. Aftab, T. Robert, and P.-B. Wieber, "Ankle, hip and stepping strategies for humanoid balance recovery with a single model predictive control scheme," in *IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 159–164.

[14] P. Kryczka, P. Kormushev, N. G. Tsagarakis, and D. G. Caldwell, "Online regeneration of bipedal walking gait pattern optimizing footstep placement and timing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 3352–3357.

[15] 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.

[16] M. R. Maximo, C. H. Ribeiro, and R. J. Afonso, "Mixed-integer programming for automatic walking step duration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 5399–5404.

[17] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "A robust walking controller based on online step location and duration optimization for bipedal locomotion," *arXiv preprint arXiv:1704.01271*, 2017.

[18] J. Ding, C. Zhou, Z. Guo, X. Xiao, and N. Tsagarakis, "Versatile reactive bipedal locomotion planning through hierarchical optimization," in *International Conference on Robotics and Automation*, 2019, pp. 256–262.

[19] J. Ding, X. Xiao, and N. G. Tsagarakis, "Nonlinear optimization of step duration and step location," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 2259–2265.

[20] H. Jeong, I. Lee, O. Sim, K. Lee, and J.-H. Oh, "A robust walking controller optimizing step position and step time that exploit advantages of footed robot," *Robotics and Autonomous Systems*, vol. 113, pp. 10–22, 2019.

[21] H. Jeong, I. Lee, J. Oh, K. K. Lee, and J.-H. Oh, "A robust walking controller based on online optimization of ankle, hip, and stepping strategies," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1367–1386, 2019.

[22] F. Nazemi, A. Yousefi-Koma, M. Khadiv *et al.*, "A reactive and efficient walking pattern generator for robust bipedal locomotion," in *RSI International Conference on Robotics and Mechatronics*, 2017, pp. 364–369.

[23] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *IEEE International Conference on Robotics and Automation*, 1991, pp. 1405–1411.

[24] Y. Wang, J. Ding, and X. Xiao, "An adaptive feedforward control method for under-actuated bipedal walking on the compliant ground," *An adaptive feedforward control method for under-actuated bipedal walking on the compliant ground*, vol. 32, no. 1, pp. 63–77, 2017.

[25] J. Luo, Y. Fu, and S. Wang, "3d stable biped walking control and implementation on real robot," *Advanced Robotics*, vol. 31, no. 12, pp. 634–649, 2017.

[26] M. Raković, B. Borovac, J. Santos-Victor, A. Batinica, M. Nikolić, and S. Savić, "Biped walking and stairs climbing using reconfigurable adaptive motion primitives," in *IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 57–62.

[27] M. Raković, B. Borovac, M. Nikolić, and S. Savić, "Realization of biped walking in unstructured environment using motion primitives," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1318–1332, 2014.

[28] F. L. Moro, N. G. Tsagarakis, and D. G. Caldwell, "Walking in the resonance with the coman robot with trajectories based on human kinematic motion primitives (kmps)," *Autonomous Robots*, vol. 36, no. 4, pp. 331–347, 2014.

[29] C. Teixeira, L. Costa, and C. Santos, "Biped locomotion-improvement and adaptation," in *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2014, pp. 110–115.

[30] D. Luo, X. Han, Y. Ding, Y. Ma, Z. Liu, and X. Wu, "Learning push recovery for a bipedal humanoid robot with dynamical movement primitives," in *IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 1013–1019.

[31] J. Pankert, L. Kaul, and T. Asfour, "Learning efficient omni-directional capture stepping for humanoid robots from human motion and simulation data," in *IEEE-RAS International Conference on Humanoid Robots*, 2018, pp. 431–434.

[32] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Non-parametric imitation learning of robot motor skills," in *International Conference on Robotics and Automation*, 2019, pp. 5266–5272.

[33] ——, "Kernelized movement primitives," *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 833–852, 2019.

[34] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[35] S. Caron, A. Kheddar, and O. Tempier, "Stair climbing stabilization of the hrp-4 humanoid robot using whole-body admittance control," in *International Conference on Robotics and Automation*, 2019, pp. 277–283.

[36] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[37] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of artificial intelligence research*, vol. 4, pp. 129–145, 1996.

[38] *database: http://mocap.cs.cmu.edu/*.

[39] L. Muratore, A. Laurenzi, E. M. Hoffman, A. Rocchi, D. G. Caldwell, and N. G. Tsagarakis, "Xbotcore: A real-time cross-robot software platform," in *IEEE International Conference on Robotic Computing*, 2017, pp. 77–80.