Robotics and Autonomous Systems 83 (2016) 287–298

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

A novel approach for the generation of complex humanoid walking sequences based on a combination of optimal control and learning of movement primitives

Debora Clever[a,*], Monika Harant[a], Henning Koch[a], Katja Mombaur[a], Dominik Endres[b]

[a] *Interdisciplinary Center for Scientific Computing (IWR), Optimization in Robotics & Biomechanics (ORB), University of Heidelberg, Germany*

[b] *Theoretical Neuroscience Group, Department Psychology, Philipps-University Marburg, Germany*

ARTICLE INFO

Article history:
Available online 21 June 2016

Keywords:
Optimal control
Movement primitives
Learning
Humanoid gait generation
HRP-2

ABSTRACT

We combine optimal control and movement primitive learning in a novel way for the fast generation of humanoid walking movements and demonstrate our approach at the example of the humanoid robot HRP-2 with 36 degrees of freedom. The present framework allows for an efficient computation of long walking sequences consisting of feasible steps of different kind: starting steps from a static posture, cyclic steps or steps with varying step lengths, and stopping motions back to a static posture. Together with appropriate sensors and high level decision strategies this approach provides an excellent basis for an adaptive walking generation on challenging terrain. Our framework comprises a movement primitive model learned from a small number of example steps that are dynamically feasible and minimize an integral mean of squared torques. These training steps are computed by solving three different kinds of optimal control problems that are restricted by the whole-body dynamics of the robot and the gait cycle. The movement primitive model decomposes the joint angles, pelvis orientation and ZMP trajectories in the example data into a small number of primitives, which effectively deals with the redundancy inherent in highly articulated motion. New steps can be composed by weighted combinations of these primitives. The mappings from step parameters to weights are learned with a Gaussian process approach, the contiguity of subsequent steps is promoted by conditioning the beginning of a new step on the end of the current one. Each step can be generated in less than a second, because the expensive optimal control computations, which take several hours per step, are shifted to the precomputational off-line phase. We validate our approach in the virtual robot simulation environment OpenHRP and study the effects of different kernels and different numbers of primitives. We show that the robot can execute long walking sequences with varying step lengths without falling, and hence that feasibility is transferred from optimized to generated motions. Furthermore, we demonstrate that the generated motions are close to torque optimality on the interior parts of the steps but have higher torques than their optimized counterparts on the steps boundaries. Having passed the validation in the robot simulator, we plan to tackle the transfer of this approach to the real platform HRP-2 as a next step.

1. Introduction

Animals in general and humans in particular are able to control, adapt and recognize the movements of their bodies seemingly without much effort, even though biological motion is a continuous dynamical process in space–time encompassing a large number of degrees of freedom. Movement primitives (MP) have been proposed as a way of simplifying this control problem, thereby facilitating the planning and execution of movements [1,2]. MPs form the link between the observable, continuous motor output and a (hypothetical) discrete internal movement representation. Another prominent hypothesis for human movement generation is based on optimal control (OC). This hypothesis posits that the human motor system is optimal (or near optimal) in its choice of movements, given the sensory input and goal specifications [3,4].

[*] Corresponding author.
E-mail address: debora.clever@iwr.uni-heidelberg.de (D. Clever).

http://dx.doi.org/10.1016/j.robot.2016.06.001
0921-8890/© 2016 Elsevier B.V. All rights reserved.

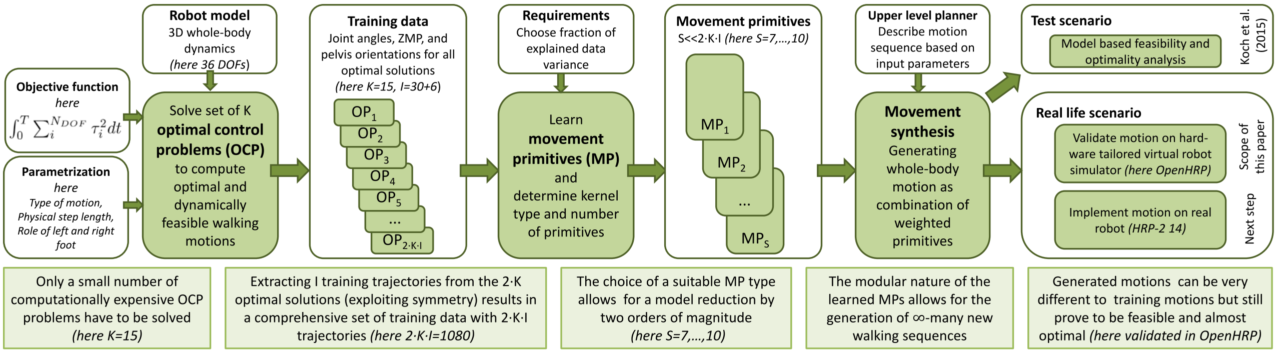
Objective function
here
$\int_0^T \sum_i^{N_{DOF}} \tau_i^2 dt$
Parametrization
here
Type of motion, Physical step length, Role of left and right foot
Robot model
3D whole-body dynamics
(here 36 DOFs)
Solve set of K optimal control problems (OCP) to compute optimal and dynamically feasible walking motions
Training data
Joint angles, ZMP, and pelvis orientations for all optimal solutions
(here K=15, I=30+6)
OP$_1$
OP$_2$
OP$_3$
OP$_4$
OP$_5$
...
OP$_{2\cdot K\cdot I}$
Requirements
Choose fraction of explained data variance
Learn movement primitives (MP) and determine kernel type and number of primitives
Movement primitives
S<<2·K·I (here S=7,…,10)
MP$_1$
MP$_2$
...
MP$_S$
Upper level planner
Describe motion sequence based on input parameters
Movement synthesis
Generating whole-body motion as combination of weighted primitives
Test scenario
Model based feasibility and optimality analysis
Koch et al. (2015)
Real life scenario
Validate motion on hard-ware tailored virtual robot simulator (here OpenHRP)
Scope of this paper
Implement motion on real robot (HRP-2 14)
Next step
Only a small number of computationally expensive OCP problems have to be solved (here K=15)
Extracting I training trajectories from the 2·K optimal solutions (exploiting symmetry) results in a comprehensive set of training data with 2·K·I trajectories (here 2·K·I=1080)
The choice of a suitable MP type allows for a model reduction by two orders of magnitude (here S=7,…,10)
The modular nature of the learned MPs allows for the generation of ∞-many new walking sequences
Generated motions can be very different to training motions but still prove to be feasible and almost optimal (here validated in OpenHRP)

Fig. 1. Methodology. Optimal and dynamically feasible motion trajectories are computed by an optimal control approach. Movement primitives are learned in a Gaussian process framework. New motions are generated by using a small number of primitives. The resulting motions prove to be sufficiently close to optimality and dynamic feasibility, validated in the virtual robot simulator OpenHRP. A corresponding video is available online: http://orb.iwr.uni-heidelberg.de/ftp/CleverEtAl_OCMP_OpenHRP (see also Appendix A).

To generate movements for humanoid robots, both approaches are very interesting. Their advantages and disadvantages can be found in two fundamentally opposite directions. Movement generation with MPs is a fast and efficient approach, applicable in the framework of on-line control. However, these primitives often model kinematics only, chosen to be safely feasible for the robot. Especially in the context of walking motions, this approach results in a conservative and generally slow motion. On the other hand, motion generation by solving optimal control problems allows for movements which exploit the limits of the robot, are dynamically feasible because they are constrained by the robot mechanics and can be optimal with respect to a variety of criteria: speed, stability, integrated torques, etc. Another benefit of OC is that it offers a natural way to solve the redundancy issue in highly articulated motions. The major drawback of this approach is the fact that it requires a high computational simulation effort and hence is only feasible for off-line use.

We aim to combine the advantages of both approaches: the fast movement generation ability of movement primitives, and the optimality and dynamical feasibility of the optimal control approach, see Fig. 1. This paper substantially extends our own previous work presented at the IEEE Humanoids conference [5]: there, we demonstrated that individual movements in form of cyclic steps based on OC can be represented in a compressed form by a simple MP model, and that novel movements can be generated by interpolation in this model. Here, we extend the MP model to allow for concatenations of steps with varying step lengths into longer walking sequences. A further novelty is the augmentation of the training data set. Whereas in [5] it exclusively consists of periodic steps with an initial velocity, here it also includes starting steps (from a static posture) and stopping steps (back to a static posture). This augmentation is essential to transfer the motions to a real robot. Furthermore, we demonstrate that the sequences thus generated can be executed by the robot (simulated by the virtual robot OpenHRP) and that the resulting movements are near-optimal most of the time. One of the most remarkable achievements of the present approach is the reduction in on-line computing time. The generation of almost optimal and feasible motions, which are executable in OpenHRP takes less than a second instead of hours (as for the pure OC based approach).

This paper is structured as follows: in Section 2, we give an overview of common MP models and OC approaches. We introduce our optimal control movement generation framework in Section 3.1, and describe the morphable MP model for novel movement production in Section 3.2. The results obtained with this combination of approaches are described in Section 4, focusing on feasibility, optimality and model comparison between MP models of different complexity. Conclusion and outlook are offered in Section 5.

2. Related work

2.1. Movement primitives

A plethora of movement primitive (MP) definitions have been proposed already. In an earlier paper [6], we argued that kinematic MPs could be grouped into temporal [7–9], spatial [10,11], and spatio-temporal primitives [2]. *Temporal* MPs are the ones which we use in this paper. They are stereotypical time-courses of degrees of freedom (DOF), e.g. joint angles. To generate complex movements, such as different human gait patterns, these time-courses are usually superimposed linearly. Modularity is an advantage of these kinematic primitives [2]: new movements can be obtained by a weighted recombination of previously learned MPs. This modularity accounts for the observed variability of human movement in the theory of MPs. It can e.g. be used to produce movements with differing styles from a small number of MPs, which has been demonstrated in computer graphics applications [12]. Movement primitives are also an important tool for efficient humanoid gait generation. In robotics, MPs are often learned from human motions and transferred to the robot (see e.g. [13]). Due to the different dynamics of human and humanoids, this approach usually is not directly applicable in the context of walking motions. Dynamical movement primitives (DMP) [14], which generate trajectories by transforming the output of a canonical dynamical system via a learned kinematics mapping onto the robot, have become popular tools for motion generation recently. DMPs can be easily modulated by task demands. The dynamical systems approach facilitates sequencing of primitives into complex actions [15]. Also, it was demonstrated in [16] that individual DMPs can encode both transient and rhythmic components of a movement. However, they are not (yet) modular (but see [17,18]), i.e. one DMP drives all relevant degrees of freedom. Thus, it remains to be shown how learned DMPs can be recombined into novel motions. Non-modular kinematic primitives were used by [19,20] for robotic gait generation. The lack of modularity results in rather large databases of MPs, containing all possible combinations of walking movements and transients which one might want to generate. This is in contrast to our modular approach, which leads to a very compact representation, as shown below.

2.2. Optimal control

Optimal control problems (OCP) restricted by multi-body dynamics are an alternative tool for off-line motion generation in humanoid robotics. They solve the redundancy issue of the underlying multi body dynamics in an elegant and beneficial way and allow to include a wide range of complicated constraints on kinematics and dynamics. Also for less complex models (e.g. the linear inverted pendulum [21]), motion generation based on OC is a beneficial approach. Beside the OC based approach itself, the solution strategy of the OCPs plays an important role, as it introduces a limit on the number of degrees of freedom that can be solved in reasonable time. Common solution methods can be classified into direct methods (direct discretization of e.g. the controls) or indirect methods, such as the Pontryagin's Maximum Principle, see e.g. [22]. Efficient solution strategies for the involved infinite dimensional state systems are based on numerical algorithms, as for example multiple shooting (e.g. [23]) or collocation methods (e.g. [24]).

A review on physics-based modeling and simulation of human walking, also including optimization-based approaches is given in [25]. Optimal control based studies on human and humanoid walking using a 3D template model are presented in [26,27]. Earlier works on whole body human and humanoid gait generation based on optimal control have been conducted by e.g. Roussel et al. in [28], Schultz and Mombaur in [29] and Erez and Todorov in [30]. The stack of tasks [31] is a whole body dynamics humanoid motion generation method with task prioritization.

Our optimal control approach goes further than many existing ones by including a detailed dynamic whole body model and the description of the gait cycle with varying contact set as constraints into the OCP.

2.3. Combining movement primitives and optimal control

As the three topics optimal control, movement primitives and learning are mostly considered by disjoint communities there is only little research on their combination, especially in the context of humanoid walking motions. In [32] a state machine based approach for humanoid gait generation using dynamic walking primitives with an optimal parametric set has been developed and successfully tested on the humanoid robot Roboray. Even though the mechanics of the system is taken into account, it is much simpler than the one used in this paper as it is modeled by lower body dynamics only. Furthermore, optimization is performed on a reduced space with less then 10 variables. Research on optimal control based movement primitives has also been done by Denk and Schmidt in [20]. Their approach is based on collocation and tailored for the humanoid Johnnie, described by 12 driven joints. However, as discussed before, those approaches are not modular and therefore require a large set of training data.

The novelty of the present approach lies in the elaborate combination of the three fields optimal control, movement primitives and learning and the use of very detailed and realistic models. This allows for a modular generation of highly dynamic walking motions with a small set of training data, that robustly get through the validation steps in the virtual robot simulator OpenHRP. Having passed those tests, there is a high probability that the motions are executable on the real platform as well.

3. The combined OC-MP framework

In the following section, we introduce the two main components of the present approach in more detail, which is the optimal control based movement generation on the one hand and the learning of a morphable movement primitive model on the other hand. As presented in Fig. 1, the present framework starts with the off-line optimal control based generation of dynamically feasible walking motions. From this training data, robot control relevant quantities are extracted and handed over to the learning process of the movement primitive model. Using a suitable number of primitives, whole body walking motions of several steps with varying step size are generated by a Gaussian Process driven morphing between primitive weights.

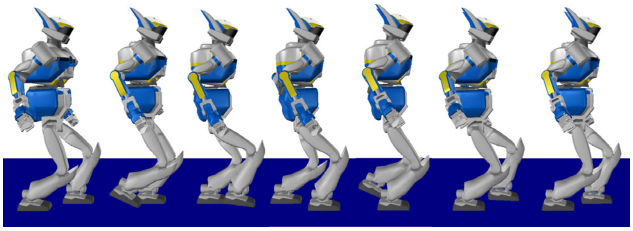

Fig. 2. Two sequential steps of a cyclic motion of constant physical step length.

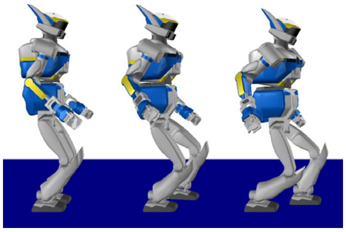

Fig. 3. Lead in motion, starting from a static posture and leading to the initial configuration of a cyclic step (see Fig. 2).

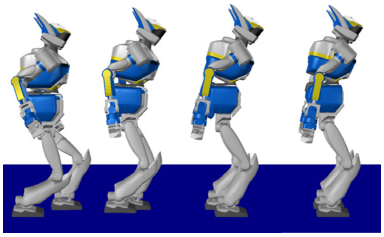

Fig. 4. Lead out motion, consisting of two sequential steps starting from a cyclic motion (see Fig. 2) and leading back to a static posture.

3.1. Optimal control based movement generation

The training data for the movement primitive mode are generated off-line as solutions of an optimal control problem (OCP). The objective of the OCPs in this example is to minimize an integral mean of the squares of the torques in the actuated joints. However, also any other objective function or combinations of objective functions could be used. The constraints of the OCP are specified by the robot's dynamics and additional restrictions that describe e.g. the gait cycle. In the present work, we formulate three different kinds of OCPs. The first OCP generates torque optimal steps of a cyclic motion with constant and predefined step length, see Fig. 2.

The second OCP generates torque optimal leading in motions, that start at the classic static posture of HRP-2 with both feet next to each other, perform one step and end at a configuration that coincides on position, velocity and force level with the initial configuration of the cyclic steps, see Fig. 3.

The third OCP generates torque optimal leading out motions, that start at the final configuration of the cyclic steps with a predefined step length (again on position, velocity and force level), perform two steps, and finally stop at a static posture with both feed next to each other, see Fig. 4.

The robot is modeled with a set of coordinates that would represent minimal coordinates in a situation without any contacts. The model consists of 36 degrees of freedom (DOF): 6 DOFs to describe the free motion of the base segment (the pelvis) in the global reference frame and 30 DOFs for its internal branched tree structure. Furthermore, we include 4 additional variables to describe the elasticity in the ankle, resulting in a sum of 40 computational variables. This coordinate configuration is preserved throughout a complete motion, where each step consists of two phases with varying contact sets (single and double support) and one discontinuity between them (touch down). Phase transitions are defined implicitly to allow for a free phase timing of the motion. The resulting equation of motion is a system of hybrid differential algebraic equations (DAE) of index three which for computational efficiency is reformulated as a system of index one.

Denoting the vector of pelvis position, pelvis orientation, joint angles and motor angles by q, the corresponding velocities by v, the acceleration by a, the joint torques by τ, the inertia matrix by M, the constraint Jacobian by G, the spatial contact constraint forces by λ, non-linear effects by NLE (e.g. Coriolis, centrifugal and gyroscopic forces), and the contact Hessian by γ, the phase-wise defined DAE is given by

$$\dot{q} = v \tag{1}$$

$$\dot{v} = a \tag{2}$$

$$\begin{pmatrix} M(q) & -G(q)^T \\ G(q) & 0 \end{pmatrix} \begin{pmatrix} a \\ \lambda \end{pmatrix} = \begin{pmatrix} -\mathrm{NLE}(q, v) + \tau \\ -\gamma(q, v) \end{pmatrix}. \tag{3}$$

Discontinuities due to inelastic contact during ground collision of a foot are defined by the system of linear equations

$$\begin{pmatrix} M(q) & -G(q)^T \\ G(q) & 0 \end{pmatrix} \begin{pmatrix} v^+ \\ \Lambda \end{pmatrix} = \begin{pmatrix} M(q)v^- \\ 0 \end{pmatrix}, \tag{4}$$

with the impulse Λ and the velocities v^- before and v^+ after the impact.

The dynamic equations are composed analytically and converted into C-code by the dynamic model builder DYNAMOD based on 6D spatial geometry [33] and symbolic code generation following [34].

To ensure physically meaningful behavior, additional constraints with respect to foot clearance, foot contact, self collision, gait cycle, hardware and control software are defined. Denoting the dynamics of the robot model by $H\dot{x} = f_j(x, u, p)$, the constraints by r_j and $c_{\mathrm{eq/ineq}}$, the state discontinuities by h_j, and the current phase of the gait cycle by $j \in \{0, \ldots, N_s - 1\}$, we formulate the gait generating OCP by

$$\min_{(x,u,p)} J(x, u, p) \tag{5}$$

$$\text{s.t.} \quad H\dot{x} = f_j(x, u, p), \tag{6}$$

$$r_j(x, u, p) \geq 0, \tag{7}$$

$$c_{\mathrm{eq}}(x(t_0), x(t_1) \ldots, x(t_f), p) = 0, \tag{8}$$

$$c_{\mathrm{ineq}}(x(t_0), x(t_1) \ldots, x(t_f), p) \geq 0, \tag{9}$$

$$x(t_j^+) = h_j(t_j^-, x(t_j^-), u(t_j^-), p), \tag{10}$$

$$j = 0, \ldots, N_s - 1.$$

For the sake of smoother torque profiles τ, the control function $u : [t_0, t_f] \to \mathbb{R}^{N_u}$ is defined by the time derivative of the torques in the actuated joints ($u := \dot{\tau}_{\mathrm{act}}, \tau := (\tau_{\mathrm{act}}, \tau_{\mathrm{pas}}), \tau_{\mathrm{pas}} \equiv 0$). The state function $x : [t_0, t_f] \to \mathbb{R}^{N_x}$ consists of pelvis translation, pelvis orientation, joint angles, all corresponding velocities and the actuated joint torques ($x := (q, v, \tau_{\mathrm{act}})$). The parameter set $p \in \mathbb{R}^{N_p}$ describes model and gait specific quantities, where some of them are free to be determined by optimization. Finally, the objective J is defined as

$$J(x, u, p) = J(q, v, \tau_{\mathrm{act}}, \dot{\tau}_{\mathrm{act}}, p) = \int_0^{t_f} \sum_{i=0}^{N_{\mathrm{DOF}}} \tau_{\mathrm{act},i}^2 \,\mathrm{d}t + \epsilon J_{\mathrm{reg}}, \tag{11}$$

to solve the redundancy issue with respect to torque minimization in the powered joints. The small regularization J_{reg} is not a regularization in the strict mathematical sense but rather a collection of terms taking into account inherent constraints imposed by the high level control system of the robot. This formulation guarantees feasibility of the generated motion on the real robot. For more details, we refer to [35].

For one step of cyclic walking motions and the lead in motions, the hybrid DAE restricting the corresponding OCP consists of two phases (single and double support) and one discontinuity between them ($N_s = 3$). For the lead out motions, which consist of two steps, the hybrid DAE restricting the corresponding OCP consists of four phases and two discontinuities after the first and the third phase ($N_s = 6$).

To sum up, this results in an optimal control problem with 110 state and 30 control functions. We solve the resulting systems with a direct approach that is based on a control discretization with local support functions, a state parameterization by multiple shooting and a structure exploiting SQP method [36], implemented in the software package MUSCOD-II [37,38]. For more details on the dynamic optimal control model of HRP-2, we refer to [23,35]. Technical information about the robot is given in [39].

As the interface to the robot platform HRP-2 takes joint angles, pelvis orientation and ZMP trajectories as input, we extract those trajectories from the OCP solutions and include them into the training data. We consider a parameterization of the different solutions of each of the three OCPs by the physical length of the step. Exploiting the symmetry of the robot, we solve the involved OCPs only for steps where the first support is on the right leg and mirror the computed solutions for the corresponding OCPs with exchanged role of legs. In the present work, we precompute optimal motions for five different step lengths (150 mm, 250 mm, 300 mm, 350 mm, 400 mm) for each of the three different motion types. Considering right and left steps, the generated training data consists of two times 15 torque optimal and dynamically feasible motions, each defined by 30 joint angle trajectories, 3 ZMP trajectories and 3 trajectories describing the pelvis orientation. This sums up to a total number of $2 \cdot 15 \cdot (30 + 3 + 3) = 1080$ training trajectories.

3.2. The morphable movement primitive model

We learn temporal MPs drawn from a Gaussian Process (GP) [40], because they are relatively easy to compute and have controllable smoothness properties. The extra modeling power of anechoic models [8] is not needed, since we generate the training data so that it is temporally aligned to the beginning of a step. Introducing variable delays between MPs might become interesting when e.g. the upper and lower body engage in different movements, such as reaching while walking [9], where anechoic MPs were shown to be superior to synchronous temporal ones. The smoothness provided by GPs facilitates interpolation to step parameter values to which the MP model has not been exposed to during training. Furthermore, we have shown previously [6] that such GP-MPs are good models for biological gait data, so we hypothesized that they might be suitable for humanoid robots, too. The graphical model of our approach is shown in Fig. 5, using plate notation: we would like to generate K movements, each with possibly different parameters l_k, e.g. step size and step type. For each movement, we need to generate I time series of movement-related signals $X_{k,i}(t)$, which in our scenario are joint angles, ZMP coordinates and global pelvis rotation angles. These signals are drawn from Gaussian distributions with standard deviations σ_i and means that are generated by linearly combining S many MPs $Y_s(t)$ with weights $W_{i,s}(l_k)$. The weights are influenced by the step parameters l_k. Thus

$$P\left(X_{k,i}(t)\right) = \mathcal{N}\left(\sum_{s=1}^{S} W_{i,s}(l_k) Y_s(t), \sigma_i^2\right). \tag{12}$$

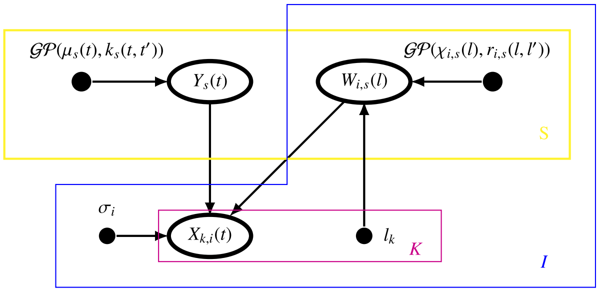
$\mathcal{GP}(\mu_s(t), k_s(t,t'))$
$Y_s(t)$
$W_{i,s}(l)$
$\mathcal{GP}(\chi_{i,s}(l), r_{i,s}(l,l'))$
S
σ_i
$X_{k,i}(t)$
l_k
K
I

Fig. 5. The morphable movement primitive model. The model generates time series $X_{k,i}(t)$ of I observable signals by multiplying S many movement primitives (MP) $Y_s(t)$ with weights $W_{i,s}$. There is one weight per signal and primitive. Weights are controlled by step parameters l_k, of which there is one instance per trial K. $\mathcal{GP}(.,.)$ indicates a Gaussian process.

As mentioned above, each MP is drawn form a GP. A GP is a prior distribution on functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$, where $d = 1$ in our application. The GP for an MP is characterized by a mean function $\mu(t)$ and one kernel function $k_s(t, t')$ per MP. The latter measures the covariance between function values at time points t and t': $\mathrm{Cov}\left[Y_s(t), Y_s(t')\right] = k_s\left(t, t'\right)$. We model the MPs in discrete time. Let $\mathbf{t} = (t_1, \ldots, t_T)$ the tuple of time points where we evaluate the MPs (and signals $X_{k,i}(t)$), and denote by $\mathbf{Y}_s = (Y_s(t_1), \ldots, Y_s(t_T))^T$ the vector of MP values at these time points. The mean vector is given by $\boldsymbol{\mu}_s = (\mu_s(t_1), \ldots, \mu_s(t_T))^T$, the $T \times T$ kernel matrix $\mathbf{K}_s$ has entries $(\mathbf{K}_s)_{i,j} = k_s(t_i, t_j)$. We use radial basis function (RBF) kernels [40] for the MPs ($\alpha, \beta, \gamma > 0$):

$$k_s(t, t') = \alpha \exp\left(-\beta |t - t'|^2\right) + \gamma \delta_{t,t'} \tag{13}$$

which give rise to smooth, non-linear functions with high probability if the GP's mean function is smooth and γ is small. We choose $\mu_s(t) = const$.

To model the weights, let $l = (\theta_1, \theta_2)$ where θ_1 is the step type and θ_2 is the step length. We draw the weight functions $W_{i,s}(l)$ from GPs with mean functions $\mu_{i,s}(l) = const$. and experimented with the RBF, linear and quadratic kernels [40]. Because the MPs and the weights interact by a product in the likelihood (Eq. (12)), an exact computation of the posterior processes is intractable. We therefore resort to approximate Bayesian inference using the variational free energy approximation [41]. Briefly, given a likelihood model $P(X|H)$ for observations X and latent variables or parameters H and a prior $P(H)$, this approximation scheme requires the choice of a (tractable) approximate posterior $Q(H)$ which is obtained by maximizing a lower bound on the marginal log-probability of the data

$$\begin{aligned} \mathcal{L} &= \langle \log\left(P(X|H)\right)\rangle_{Q(H)} - D\left(Q(H) \parallel P(H)\right) \\ &\leq \log(P(X)) \end{aligned} \tag{14}$$

where the angle brackets denote an expectation under $Q(H)$, and $D\left(Q(H) \parallel P(H)\right) = \langle \log(Q(H))\rangle_{Q(H)} - \langle \log(P(H))\rangle_{Q(H)}$ is the Kullback–Leibler divergence between posterior and prior [42]. This approximation scheme has several appealing features (see [41]), the most important one for our purposes is that $\mathcal{L}$ can be used for approximate model comparison, which we employ to find the best number of MPs. To apply this approximation to our data, assume we observed a tensor of signals $\mathbf{X}$ with $\mathbf{X}_{k,i,n} = X_{k,i}(t_n)$ for all $t_n \in \mathbf{t}$ and associated step parameters l_k. Let $\mathbf{W}_{i,s} = (W_{i,s}(l_1), \ldots, W_{i,s}(l_K))^T$ be the weight vector for signal i and MP s, $\boldsymbol{\chi}_{i,s} = (\chi_{i,s}(l_1), \ldots, \chi_{i,s}(l_K))^T$ the associated mean vector and $\mathbf{R}_{i,s}$ the kernel matrices for these weights. Using slice notation (e.g. $\mathbf{Y}_{:}$=$(\mathbf{Y}_1, \ldots, \mathbf{Y}_S)$), the MP model can then be expressed in discrete time as

$$P(\mathbf{X}) = \prod_{k,i,n} \mathcal{N}\left(\sum_s (\mathbf{W}_{i,s})_k (\mathbf{Y}_s)_n, \sigma_i^2\right) \tag{15}$$

$$P(\mathbf{W}_{:,:}) = \prod_{i,s} \mathcal{N}\left(\boldsymbol{\chi}_{i,s}, \mathbf{R}_{i,s}\right) \tag{16}$$

$$P(\mathbf{Y}_{:}) = \prod_{s} \mathcal{N}\left(\boldsymbol{\mu}_s, \mathbf{K}_s\right). \tag{17}$$

The parameters/latent variables H in Eq. (14) are therefore comprised of $\mathbf{W}_{:,:}$ and $\mathbf{Y}_{:}$ in our model. For the posterior approximation, we assume that the distributions of weights and MPs are conjugate to the prior (i.e. multivariate Gaussians, Eqs. (16) and (17)) and factorize between weights and MPs, $Q\left(\mathbf{W}_{:,:}, \mathbf{Y}_{:}\right) = Q\left(\mathbf{W}_{:,:}\right)Q\left(\mathbf{Y}_{:}\right)$. Posterior parameters are in the following denoted with a tilde, e.g. $\tilde{\boldsymbol{\chi}}_{i,s}$ is the posterior mean of the weight that connects signal i with MP s. With these choices, it is straightforward (but tedious) to evaluate the variational lower bound $\mathcal{L}$ (Eq. (14)). Using Eq. (15) and the conjugacy of the posterior, we find for the expected log-likelihood

$$\begin{aligned} \langle P(\mathbf{X})\rangle_Q = &-\frac{K \cdot T}{2}\left(I \cdot \log(2\pi) + \sum_i \log(\sigma_i^2)\right) \\ &- \sum_{k,i,n} \frac{1}{2\sigma_i^2}\left(\mathbf{X}_{k,i,n} - \sum_s (\tilde{\boldsymbol{\chi}}_{i,s})_k (\tilde{\boldsymbol{\mu}}_s)_n\right)^2 \\ &- \frac{1}{2}\sum_s \left(\mathrm{tr}\left[\tilde{\mathbf{K}}_s\right] \sum_{i,k} \frac{(\boldsymbol{\chi}_{i,s})_k^2}{\sigma_i^2}\right. \\ &+ \sum_i \frac{\mathrm{tr}\left[\tilde{\mathbf{R}}_{i,s}\right]}{\sigma_i^2} \sum_n (\boldsymbol{\mu}_s)_n^2 \\ &\left. + \mathrm{tr}\left[\tilde{\mathbf{K}}_s\right] \sum_i \frac{\mathrm{tr}\left[\tilde{\mathbf{R}}_{i,s}\right]}{\sigma_i^2}\right). \end{aligned} \tag{18}$$

The first term on the right hand side results from the normalization constant of the Gaussian, the second term measures the squared deviation between the data and the expectations of weights and MPs, and the third term leads to a preference for small posterior variances and means. The second term required for the evaluation of $\mathcal{L}$ is given by the well-known formula for the Kullback–Leibler divergence between multivariate Gaussians [43]:

$$\begin{aligned} &D(\mathcal{N}(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{K}}) \parallel \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})) \\ &\quad = \frac{1}{2}\left(\mathrm{tr}\left[\mathbf{K}^{-1}\tilde{\mathbf{K}}\right] + (\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})^T \mathbf{K}^{-1}(\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})\right. \\ &\quad \left. - \dim[\boldsymbol{\mu}] + \log(|\mathbf{K}|) - \log(|\tilde{\mathbf{K}}|)\right). \end{aligned} \tag{19}$$

In summary, the variational lower bound for our model is given by

$$\begin{aligned} \mathcal{L} = &\langle P(\mathbf{X})\rangle_Q - \sum_s D(\mathcal{N}(\tilde{\boldsymbol{\mu}}_s, \tilde{\mathbf{K}}_s) \parallel \mathcal{N}(\boldsymbol{\mu}_s, \mathbf{K}_s)) \\ &- \sum_{i,s} D(\mathcal{N}(\tilde{\boldsymbol{\chi}}_{i,s}, \tilde{\mathbf{R}}_{i,s}) \parallel \mathcal{N}(\boldsymbol{\chi}_{i,s}, \mathbf{R}_{i,s})). \end{aligned} \tag{20}$$

We learn the model from $\mathbf{X}$ by maximizing $\mathcal{L}$ with respect to the posterior parameters $\tilde{\boldsymbol{\mu}}_{:}, \tilde{\mathbf{K}}_{:}, \tilde{\boldsymbol{\chi}}_{:,:}, \tilde{\mathbf{R}}_{:,:}$ and the kernel parameters α, β, γ. To this end, we use the limited memory Broyden–Fletcher–Goldfarb–Shanno algorithm for constrained optimization [44] implemented in the SciPy package [45]. Automatic gradient computation is done with the Theano library [46]. Since $\mathcal{L}$ is not globally convex, we optimize one group of variables at a time, and iterate this procedure until convergence. We compute a starting point for the optimization by a singular value decomposition (SVD) of the data, like we did in [5]. All noise variances σ_i^2 are set to 0.01 times the corresponding signal variance, i.e. we aim for 0.99 variance accounted for (VAF). This level of VAF is necessary for stable walking motion, see results below. Prior means are initialized to the average of the posterior means, kernel parameters are initialized by optimization of the standard marginal probability of a GP [40], given then p(oste)rior means. There is one set of kernel parameters per MP and weight function (cf. Fig. 5).

3.2.1. Generating new movements

Once the model has been learned, we use it to generate new movements. These movements need to fulfill three conditions: first, they should obey, as closely as possible, a sequence of provided step parameters. Second, they need to be contiguous so that the robot can execute them. Third, generation should run in real time. Fulfillment of the first condition is promoted by generating weights from the posterior distribution inferred during the learning process. The second condition is met by conditioning the beginning of the movement currently generated on the end of the previous movement (if such exists). We can rephrase these two conditions by restating inference as an optimization problem, as we did for learning. To make this optimization problem tractable in (near) real-time, we can only afford to optimize with respect to a small set of parameters. We therefore decided to keep the posterior distributions of the MPs and the weights found during learning fixed, and optimize only w.r.t. the weight posterior distribution of the new movement. Denote the weight of source s for signal i of the new movement by $\hat{W}_{i,s}$. We assume that the joint variational posterior of the weights is multivariate Gaussian and factorize it as

$$\begin{aligned} Q(\hat{W}_{:,:}, \mathbf{W}_{:,:}) &= \prod_{i,s} Q(\hat{W}_{i,s}, \mathbf{W}_{i,s}) \\ &= \prod_{i,s} Q(\hat{W}_{i,s}|\mathbf{W}_{i,s}) Q(\mathbf{W}_{i,s}). \end{aligned} \tag{21}$$

Since $Q(\mathbf{W}_{i,s})$ is a multivariate Gaussian parameterized by $\tilde{\boldsymbol{\chi}}_{i,s}$, $\tilde{\mathbf{R}}_{i,s}$, a natural choice for $Q(\hat{W}_{i,s}|\mathbf{W}_{i,s})$ is a conditional Gaussian:

$$Q(\hat{W}_{i,s}|\mathbf{W}_{i,s}) = \mathcal{N}\left(\hat{\chi}_{i,s} + \hat{\mathbf{M}}_{i,s}\mathbf{W}_{i,s}, \hat{R}_{i,s}\right) \tag{22}$$

where the mean $\hat{\chi}_{i,s} + \hat{\mathbf{M}}_{i,s}\mathbf{W}_{i,s}$ is a generalized form of the mean of a conditional Gaussian (see [43], eqn. (353)) and $\hat{R}_{i,s}$ is the variance. Note that this expression connects the new $\hat{W}_{i,s}$ to the previous $\mathbf{W}_{i,s}$ whose distribution was determined during learning. The variational posterior parameters of the new weight's distribution are therefore $\hat{\chi}_{i,s}$, $\hat{\mathbf{M}}_{i,s}$ and $\hat{R}_{i,s}$, with respect to which we maximize the variational free energy. To do so, we exploit the fact that the prior of $\mathbf{W}_{i,s}$ is a conditional Gaussian, because the weights are generated by functions drawn from a Gaussian process. By virtue of eqn. (353) in [43] and the above definition of the kernel matrices, the conditional prior mean and variance can be written as follows: denote the parameter of the new step to generate with l' and let the vector $\mathbf{Z}_{i,s} = (r_{i,s}(l', l_1), \ldots, r_{i,s}(l', l_K))^T$. Then the conditional prior mean and variance are given by

$$m_{i,s} = \chi_{i,s}(l') + \mathbf{Z}_{i,s}^T \mathbf{R}_{i,s}^{-1} (\mathbf{W}_{i,s} - \boldsymbol{\chi}_{i,s}) \tag{23}$$

$$R_{i,s} = r_{i,s}(l', l') - \mathbf{Z}_{i,s}^T \mathbf{R}_{i,s}^{-1} \mathbf{Z}. \tag{24}$$

Furthermore, note that the Kullback–Leibler divergence between weight posterior and prior can be decomposed as

$$\begin{aligned} &D(Q(\hat{W}_{i,s}, \mathbf{W}_{i,s}) \parallel P(\hat{W}_{i,s}, \mathbf{W}_{i,s})) \\ &\quad = \left\langle D(Q(\hat{W}_{i,s}|\mathbf{W}_{i,s}) \parallel P(\hat{W}_{i,s}|\mathbf{W}_{i,s})) \right\rangle_{Q(\mathbf{W}_{i,s})} \\ &\qquad + D(Q(\mathbf{W}_{i,s}) \parallel P(\mathbf{W}_{i,s})) \end{aligned} \tag{25}$$

which follows directly from its definition. The second term on the right hand side is fixed during movement generation, because we only optimize with respect to the parameters of $Q(\hat{W}_{i,s}|\mathbf{W}_{i,s})$. For the assumed form of the approximating posterior, we define

$$\begin{aligned} \Delta\mathbf{M}_{i,s} &= \hat{\mathbf{M}}_{i,s} - \mathbf{M}_{i,s} \\ \Delta\chi_{i,s} &= \hat{\chi}_{i,s} + \hat{\mathbf{M}}_{i,s}\tilde{\boldsymbol{\chi}}_{i,s} - \chi_{i,s}(l') - \mathbf{M}_{i,s}(\tilde{\boldsymbol{\chi}}_{i,s} - \boldsymbol{\chi}_{i,s}) \end{aligned}$$

and find

$$\begin{aligned} &\left\langle D(Q(\hat{W}_{i,s}|\mathbf{W}_{i,s}) \parallel P(\hat{W}_{i,s}|\mathbf{W}_{i,s})) \right\rangle_{Q(\mathbf{W}_{i,s})} \\ &\quad = \frac{1}{2}\left(R_{i,s}^{-1}\hat{R}_{i,s} + \Delta\chi_{i,s}^2 R_{i,s}^{-1} + \log\left(\frac{R_{i,s}}{\hat{R}_{i,s}}\right) \right. \\ &\qquad \left. - 1 + \Delta\mathbf{M}_{i,s}^T R_{i,s}^{-1} \Delta\mathbf{M}_{i,s}\hat{R}_{i,s} \right) \end{aligned} \tag{26}$$

where $\mathbf{M}_{i,s} = \mathbf{Z}_{i,s}\mathbf{R}_{i,s}^{-1}$.

We similarly decompose the likelihood component of the variational free energy into a part that depends on the $\hat{W}_{:,:}$, and a part that does not. The first movement in a sequence has no $\hat{W}_{:,:}$-dependent likelihood component, because there is no previous movement to which it must be connected. For all others, it can be obtained from Eq. (18) by substituting $\mathbf{X}_{k,i,s}$ with the end of the previous movement, and the posterior parameters of the training movements with the posterior parameters of the new movement.

4. Transfer of motions to virtual robot

In this section, we use the developed OC-MP framework to generate walking sequences for the humanoid robot HRP-2. The resulting motions are verified by the virtual robot simulator OpenHRP, which is also the interface to the real robot. While the transfer of motions to the real robot is the next (immediately following) step, this paper is devoted to a thorough analysis of the quality of generated motions and their transferability.

The section is divided into three parts. In the first and second part we consider walking sequences which consist of a starting step, two steps of constant step length, and two stopping steps. We focus on feasibility issues (part one) and quality issues (part two), to answer the following questions:

- Is it possible to generate feasible walking motions with steps of a certain length that have not been part of the training data?
- What is a suitable kernel and how many primitives are necessary to describe the walking motions under investigation?
- How well does the resulting step length (based on the generated joint angles) correlate to the desired step length?
- Based on torque optimal training data, how optimal are the generated motions with respect to an integral mean of the sum of squared torques?

In the third part, we focus on the transferability of the generated motions to the robot. To this end, we consider motions that consist of several steps of varying step length, to answer the following questions:

- Is it possible to generate transitions between steps of different physical step length leading to feasible walking motions?
- If yes, how do those transitions look like and does this increase the number of necessary movement primitives?
- Are the resulting contact forces in the range of the robot bounds?

Table 1
Relation between kernel type, number of MPs and feasibility. A motion is defined to be feasible if the walking sequence $(1+2+2$ steps) can be performed by the virtual robot simulator OpenHRP without a fall. The numbers in the table indicate how many of the 10 performed trials (with different step lengths) have been shown to be feasible.

#MPs	LIN	QUAD	RBF
2	0	0	0
3	0	0	0
4	0	0	3
5	2	4	6
6	9	9	10
7	10	10	10
8	10	9	10

4.1. Feasibility of learned motions

To investigate an appropriate kernel and a suitable number of movement primitives, we generate 210 different walking sequences, which consist of a starting step on the right leg, two steps of constant step length (first stance foot left, second right), and two stopping steps (stance left, stance right). Next to the five different step lengths that have been considered in the training data (150 mm, 250 mm, 300 mm, 350 mm, 400 mm), we investigate the behavior of motions with intermediate step length 175 mm, 225 mm, 275 mm, 325 mm, and 375 mm. For each of the 10 motions, we investigate three different kinds of kernels for the GPs: a linear (LIN) one, a quadratic (QUAD) one, and a radial basis function (RBF) kernel. For each kernel and each step length, we generate motions that are based on two to eight MPs, inspired by our previous work, where we found that five MPs are enough to approximate the cyclic steps [5]. The feasibility of the generated motions (which are described by joint angles, ZMP trajectories and pelvis orientation) is verified by the virtual robot simulator OpenHRP, that reliably simulates the behavior to be expected on the real platform. We consider a motion to be feasible if it can be performed by the robot without falling. The number of feasible motions out of 10 (with different step lengths) for each number of MPs and each of the three different kernels is presented in Table 1.

We find that for very small numbers of primitives (2–3) none of the three approaches leads to feasible motions. The RBF-kernel based approach works more reliably for smaller numbers of primitives (4–5) than the others. In these cases (4–5 primitives), failure is not related to motions with interpolated step lengths but to motions with longer step lengths as those are very dynamical. Based on the results in Table 1, we suggest that seven primitives with RBF-kernel are suitable to describe the motions of interest.

An example for the transition between the lead in motion and the first step for optimized and generated motions (also for interpolated step lengths) is presented in Fig. 6.

In addition to the analysis performed above, we evaluate the variational free energy $\mathcal{L}$ and the variance accounted for (VAF) in Fig. 7 for the joint angle trajectories. VAF is the fraction of the variance of the training data which can be explained by a MP model with a given number of primitives. For all three kernels, the free energy increases sharply until $\approx$6 primitives, reaching its maximum at 6/9/11 MPs for the quadratic/linear/RBF kernel, respectively. Note that these maxima are rather flat, indicating that a range of MPs should perform similarly well. These observations are consistent with the feasibility results in Table 1: at $\geq$6 MPs, the number of feasible movements could not increase further. For the RBF kernel, which performs best in terms of $\mathcal{L}$ and VAF, the model comparison based on $\mathcal{L}$ predicts that $\approx$11 MPs should have a slight advantage over 7 MPs (see also inset in Fig. 7, left). This prediction is supported by our experiments with long walking sequences below. VAF expectably keeps increasing with the number of MPs, except for the quadratic kernel which shows signs of numerical instability at higher number of MPs, indicating that the quadratic kernel is a poor choice for this application. The superior performance of the RBF kernel, both in the feasibility study (Table 1) and in the approximate model comparison, indicates a smooth nonlinear relationship between step parameters and weights that cannot be modeled with linear or quadratic kernels.

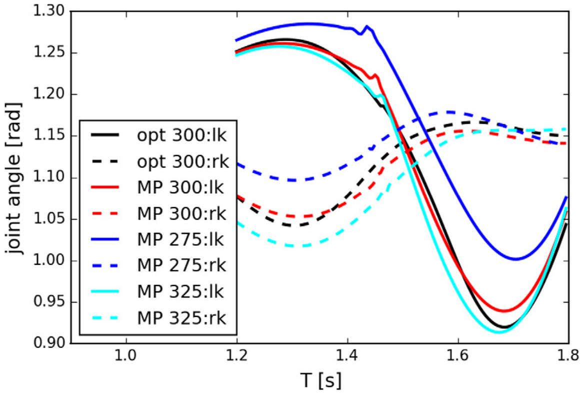

Fig. 6. Comparison of transition from lead-in to step, between optimal control (opt) and MP-based trajectories for the knee flexion angle. Numbers are planned step lengths, lk/rk: left/right knee. The transition point is at $T \approx 1.45$ s. Note, that the figure only shows the transition phase and not the complete lead in motion before. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

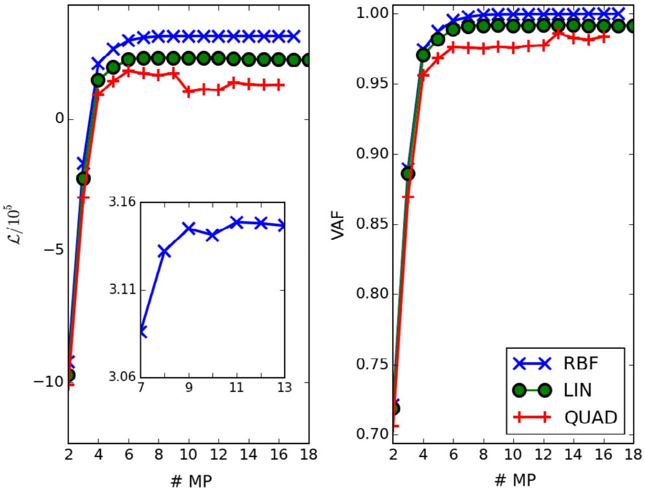

Fig. 7. Model comparison results. *Left*: variational free energy $\mathcal{L}$ for different numbers of MPs and weight morphing kernels (RBF, linear, quadratic). Inset shows $\mathcal{L}$ for the RBF kernel around its maximum at 11 MPs. *Right*: variance accounted for (VAF) as a function of the number of MPs, for the same three weight morphing kernels as on the left. In line with the experimental results in OpenHRP, the RBF kernel performs best. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.2. Quality of learned motions

Based on the results of the previous subsection, we now focus on motions of the above described type, that are generated with the RBF-based approach and seven movement primitives. To study how close to optimality these motions are, we are interested in the quality of the resulting physical step length with respect to the desired one and in the resulting integral mean of the sums of squares of the joint torques, which is the relevant part of the objective that the training data has been optimized for.

As the physical step length is not an explicit state of the model but a result of the concatenation of joint angles, it is not even

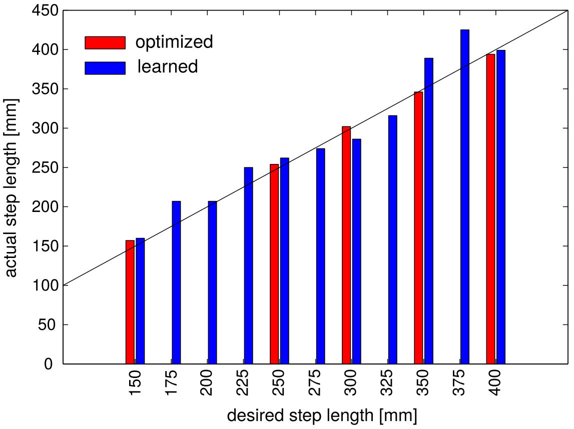

Fig. 8. Resulting physical step lengths for computed motions when tested with OpenHRP. Results of pure OC based motions (training data) are plotted in red, those of the MP-generated motions (also for interpolated step lengths) in blue. In the ideal case, the upper right corner of the red bars and the upper left corner of the blue bars intersect with the bisectrix (black). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

clear, if the optimized motions (training data) result in the desired step length, when being tested in the virtual robot simulator. Even more, it is unclear, how well the desired step length is met for motions with interpolated step length that are based on a GP morphing between training examples. In Fig. 8, we present the resulting step lengths of the considered motions, when these motions are tested in the robot simulator. The resulting step lengths of motions computed with optimal control are plotted in red, those resulting from the generated motions (based on the learned primitives) in blue. Already for the small number of seven primitives, deviations are between zero and five centimeters, which is good enough for many applications. If higher precision is needed (e.g. for a complicated step stone scenario), an increase of the number of movement primitives is a promising option.

As all motions that have been included in the training data mainly minimize the integral mean of the sums of squares of the joint torques

$$J = \int_0^{t_f} \sum_{i=0}^{N_{\mathrm{DOF}}} \tau_i^2 \mathrm{d}t,$$

it is an important question how well this kind of optimality transfers to the generated motions—also those with interpolated step length. Whereas the torques of the generated motions basically coincide with the optimized motions during the steps, on the step boundaries (between two steps) the generated motions result in peak torques, that are increased approximately by a factor of five. To illustrate this behavior, we split the motion into the interior parts of a step which take the main portion of the motion (between 0.75 s and more than 2 s) and the boundary parts, which are significantly shorter (between 0.05 s and 0.25 s). Neglecting the regularization (J_{reg}, Eq. (11)), in Fig. 9 the resulting objective values for motions generated based on OC are plotted in red, and the objective values for motions generated based on MPs in blue. For the interior part (top), we observe a marginal increase of mean torques for generated steps in comparison to their optimized counterparts when the motion is tested in OpenHRP. For interpolated step lengths optimality is almost achieved. This slight increase of mean torque correlates with the distance of physical step length to the closest one contained in the training data. In contrast, the great increase of mean torque for MP-generated motions on the step boundaries (bottom of Fig. 9) occurs for both training and interpolated step lengths. So far, we have studied if an increase of movement primitives would solve the problem. As this is not the case, this effect seems to be a result of a sudden compensation strategy of OpenHRP and further investigations on its trigger are necessary.

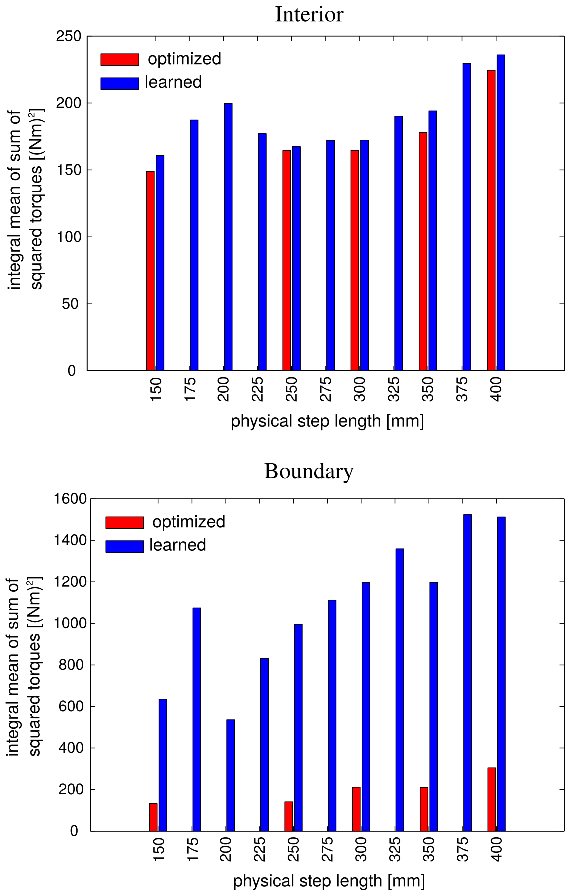

Fig. 9. Resulting integral mean of sum of squared torques (objective) for computed motions when tested with OpenHRP. *Top*: Values for interior part of steps. We observe a marginal increase of mean torques for MP-generated steps in comparison to their OC-based counterparts. For interpolated step lengths, optimality is almost achieved. *Bottom*: Values on the boundary of steps. Independently of the fact if we consider a step length contained in the training data or an interpolated step length, on step boundaries the mean torques of MP-generated motions increase approximately by a factor of five. Note, that the mean value of the torques themselves is given by the root of the present values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

However, taking torque limits into account, all considered motions are feasible for the robot simulator and hence are likely to be feasible for the real platform as well.

4.3. Transfer between different steps

In this section, we investigate concatenations of steps with varying step length (between 150 mm and 400 mm). As the cyclic motions from the training data do not include transitions between steps of different physical step length, this is a pure result of the learning process and a great achievement for the fast generation of dynamically feasible walking motions.

We study the behavior of three different walking motions, which consist of 1+6+2 sequential steps of the following step lengths:

- Motion 1: 225 mm (lead in), 225 mm, 250 mm, 275 mm, 300 mm, 275 mm, 250 mm, 250 mm (lead out),
- Motion 2: 225 mm (lead in), 250 mm, 275 mm, 300 mm, 325 mm, 350 mm, 375 mm, 400 mm (lead out),

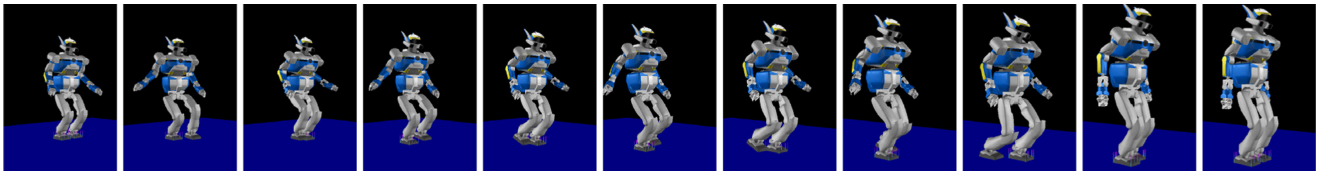

Fig. 10. Generated walking sequence (Motion 1), validated in virtual robot simulator OpenHRP. The screen shots are captured with a frequency of 1 Hz. The purple arrows on the feet visualize the contact forces. A corresponding video is available online: http://orb.iwr.uni-heidelberg.de/ftp/CleverEtAl_OCMP_OpenHRP (see also Appendix A). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

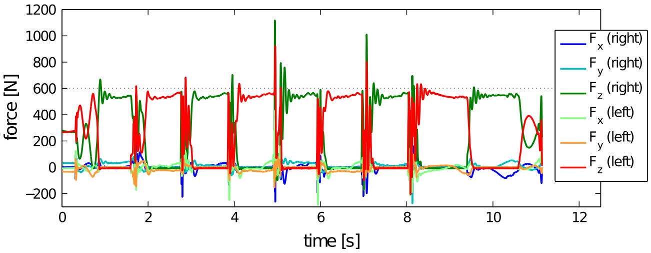

Fig. 11. Resulting contact forces (motion 1) when tested in OpenHRP. The resulting contact forces are below the guideline of 600 N most of the time. It is expected that the short violations of this guideline do not cause major problems for the transfer to the real robot.

- Motion 3: 150 mm (lead in), 175 mm, 200 mm, 225 mm, 250 mm, 225 mm, 200 mm, 200 mm (lead out).

Finally, we also consider a very long motion, which consist of 1+20+2 sequential steps of the following step lengths:

- Motion 4: 150 mm (lead in), 175 mm, 200 mm, 225 mm, 250 mm, 275 mm, 300 mm, 325 mm, 350 mm, 375 mm, 400 mm, 375 mm, 350 mm, 325 mm, 300 mm, 275 mm, 250 mm, 225 mm, 200 mm, 175 mm, 150 mm, 150 mm (lead out).

The first motion represents a symmetric increase and decrease of step length. The second motion spans a wide range of step lengths, in particular the ones with large step sizes, which are the most dynamic ones. The third sequence includes mainly step lengths that are not part of the training data. They are located in the gap between 150 and 250 mm. The last motion is by far the most challenging, as it includes step lengths of the entire range of physical step lengths and a great variety of different transitions between them. Note, that the step length parameter of the lead out motion has to be equal or greater than the last step. Lead out motions with a step length parameter smaller than the last step cannot capture the forward motion of the robot and cause a shaking or even a tipping over.

Using the RBF kernel and seven movement primitives, the generation attempts for the three shorter sequences (motion 1–3) all result in a feasible walking motion, that can be executed by the virtual robot simulator without a fall. See Fig. 10 for a sequence of screenshots of motion 1 (videos for all three motions are available on-line, see Appendix A). The purple arrows on the feet visualize the contact forces. It is preferable that the contact forces do not exceed the guideline of 600 N for longer periods of time. As can be seen in Fig. 10, the resulting contact forces are distributed quite homogeneously and Fig. 11 shows that they stay within the desired range most of the time. Therefore, we expect a high probability that the motion is suitable for the real platform as well.

To get an impression of the modifications of joint angle trajectories when concatenating steps of varying step length, in Fig. 12 we present the knee flexion angle trajectories for transitions of optimized motions between steps of equal step length (300 mm), for transitions of learned motions between steps of equal step length (300 and 325 mm), and finally for transitions of learned motions between steps of different step length (300 and 325 mm). Transitions between other steps lengths and for other joints are comparable.

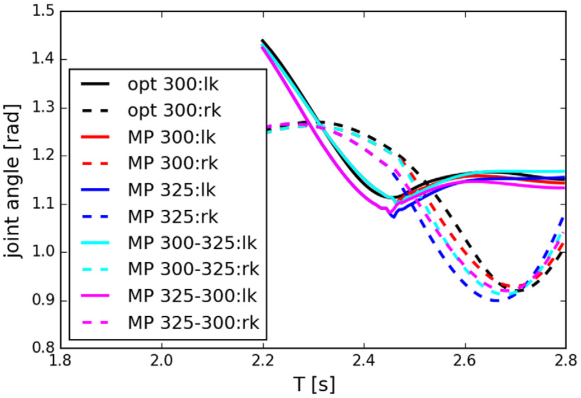

Fig. 12. Comparison of transition from step to step of same and different lengths, between optimal control (opt) and MP-based trajectories for the knee flexion angle. Numbers are planned step lengths, lk/rk: left/right knee. The transition point is at $T \approx 2.5$ s. Note, that the figure only shows the transition phase and not the complete motion before.

Finally, the robustness of the novel motion generation method and the choice of kernel type and number of primitives is validated by a very long walking sequence (Motion 4) that consists of 20 steps (excluding lead in and lead out) that vary from steps with a step length of 150 mm to steps with a step length of 400 mm and finally back to steps with a length of 150 mm. As for the choice of the RBF kernel and seven MPs, the robot falls down at the very end of the motion, we computed the motion with 10 MPs. This modification is enough to result in a feasible motion without fall, see Fig. 13. As mentioned before, this observation coincides nicely with the computational result in part one of this section, where we predicted that for the RBF kernel the quality of the motions could be further improved, by increasing the number of MPs to approximately eleven.

5. Conclusions

Our fusion of optimal control and movement primitive learning combines the best of both worlds: dynamical feasibility and optimality on the one hand, and compact representation, fast movement generation and re-use of learned MPs through modularity on the other. Besides testing our approach on the real robot, we would like to extend it in several directions: so far, we can only do open-loop control, for which temporal MPs are suitable. An advantage of DMP-like primitives with contracting [14] or partially contracting [47], canonical dynamical systems is their ability to respond on-line to perturbations and unexpected changes to sensory feedback. We therefore plan to combine the advantages of modular MPs with those of dynamical MPs, where (in contrast to the common use of DMPs) we also include the dynamics (in a mechanical sense) of the whole body system. Our first steps in this direction are promising [18,48],

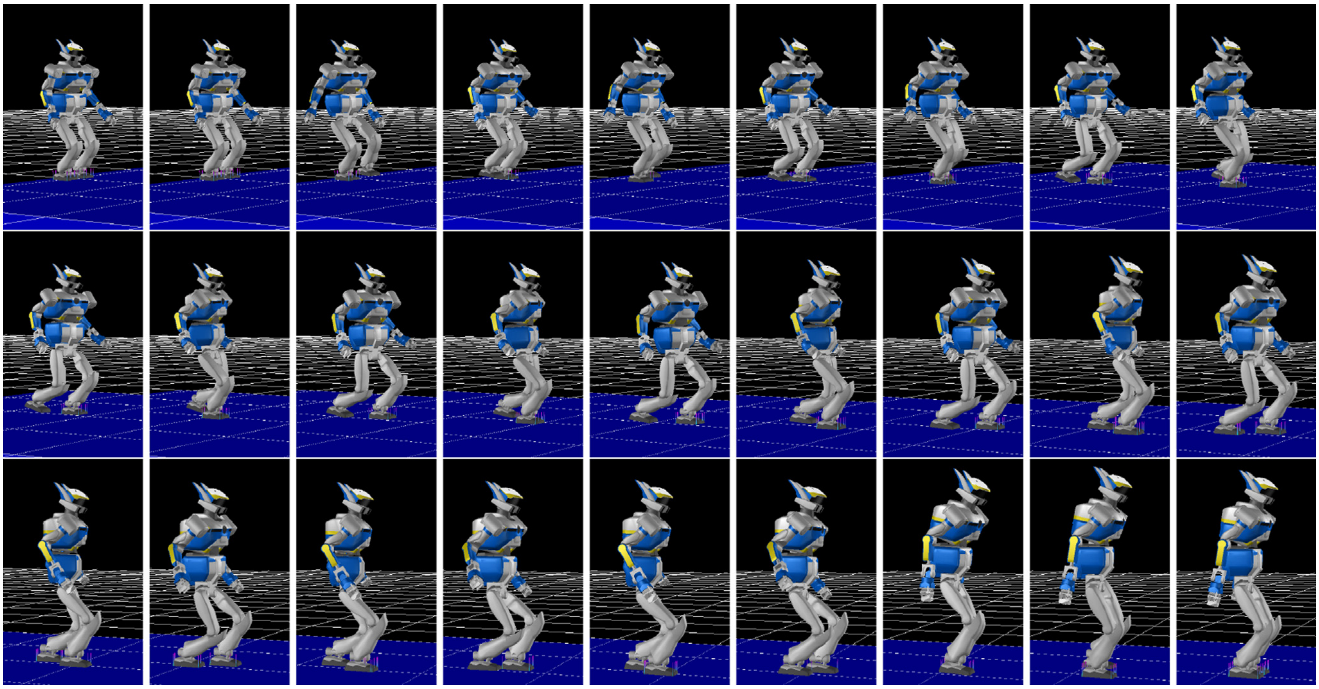

Fig. 13. Generated walking sequence with 20 + 3 steps of varying physical step length (from 150 mm to 400 mm back to 150 mm), validated in virtual robot OpenHRP. The screen shots are captured with a frequency of 1 Hz. The purple arrows on the feet visualize the contact forces. A corresponding video is available on-line (see Appendix A). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

but a demonstration of these modular dynamical MPs on large movement databases is still missing. Another extension direction is the inclusion of sensory signals for (on-line) planning, e.g. [20]. Here, it might even be useful to use kinematical and dynamical MPs in parallel: the former for their facilitation of planning, because movement end points can be determined without the need for roll-outs; the latter for the aforementioned ability to deal with perturbations during execution.

It can also be seen, in comparison to our previous work [5], that the number of necessary movement primitives does not increase significantly, even though the number of training data trajectories has been increased from 180 to 1080. Moreover, this number can be estimated with approximate Bayesian model comparison, yielding long and stable walking sequences. This motivates future plans to augment the set of training data by further motion types (e.g. curve walking) and further optimality criteria (e.g. maximal walking velocity or maximal efficiency), while still maintaining a compact MP representation of such data.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Program (FP7/2007–2013) under grant agreement no. 611909 (KoroiBot), from the HGS MathComp Graduate School, from the DFG under IRTG 1901 'The Brain in Action' and project C06 of the SFB/TRR 135 'Cardinal Mechanisms of Perception'.

We want to thank the Simulation and Optimization research group of the IWR at Heidelberg University for giving us the possibility to work with MUSCOD-II. Furthermore we thank O. Stasse (CNRS-LAAS) for his help in tailoring the problem formulation to the hardware issues of HRP-2.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.robot.2016.06.001.

References

[1] N.A. Bernstein, The Co-ordination and Regulation of Movements, Pergamon Press, Oxford, 1967.

[2] A. d'Avella, M.C. Tresch, Modularity in the motor system: decomposition of muscle patterns as combinations of time-varying synergies, in: T.G. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, Vol. 14, The MIT Press, Cambridge, MA, 2002, pp. 141–148.

[3] E. Todorov, Optimality principles in sensorimotor control, Nature Neurosci. 7 (9) (2004) 907–915. http://dx.doi.org/10.1038/nn1309.

[4] K.P. Körding, D.M. Wolpert, Bayesian decision theory in sensorimotor control, Trends Cogn. Sci. 10 (7) (2006) 319–326. http://dx.doi.org/10.1016/j.tics.2006.05.003.

[5] K.H. Koch, D. Clever, K. Mombaur, D. Endres, Learning movement primitives from optimal and dynamically feasible trajectories for humanoid walking, in: IEEE-RAS International Conference on Humanoid Robots (Humanoids 2015), pp. 866–873, 2015. http://dx.doi.org/10.1109/HUMANOIDS.2015.7363463.

[6] D.M. Endres, E. Chiovetto, M.A. Giese, Model selection for the extraction of movement primitives, Front. Comput. Neurosci. 7 (2013) 185. http://dx.doi.org/10.3389/fncom.2013.00185, URL http://www.frontiersin.org/computational_neuroscience/10.3389/fncom.2013.00185/abstract.

[7] E. Chiovetto, B. Berret, T. Pozzo, Tri-dimensional and triphasic muscle organization of whole-body pointing movements, Neuroscience 170 (4) (2010) 1223–1238. http://dx.doi.org/10.1016/j.neuroscience.2010.07.006.

[8] L. Omlor, M.A. Giese, Anechoic blind source separation using Wigner marginals, J. Mach. Learn. Res. 12 (2011) 1111–1148. URL http://jmlr.csail.mit.edu/papers/volume12/omlor11a/omlor11a.pdf.

[9] E. Chiovetto, M.A. Giese, Kinematics of the coordination of pointing during locomotion, PLoS One 8 (11) (2013) e79555. http://dx.doi.org/10.1371/journal.pone.0079555.

[10] F.A. Mussa-Ivaldi, S.A. Solla, Neural primitives for motion control, IEEE J. Ocean. Eng. 29 (3) (2004) 640–650.

[11] C.B. Hart, S.F. Giszter, Distinguishing synchronous and time varying synergies using point process interval statistics: Motor primitives in frog and rat, Front. Comput. Neurosci. 7 (52) (2013) http://dx.doi.org/10.3389/fncom.2013.00052, URL http://www.frontiersin.org/computational_neuroscience/10.3389/fncom.2013.00052/abstract.

[12] M.A. Giese, A. Mukovskiy, A.-N. Park, L. Omlor, J.-J. Slotine, Real-time synthesis of body movements based on learned primitives, in: D. Cremers, B. Rosenhahn, A.L. Yuille (Eds.), Statistical and Geometrical Approaches to Visual Motion Analysis, in: Lecture Notes in Computer Science, vol. 5604, 2009, pp. 107–127.

[13] F.L. Moro, N.G. Tsagarakis, D.G. Caldwell, A human-like walking for the compliant humanoid coman based on com trajectory reconstruction from kinematic motion primitives, in: IEEE-RAS International Conference on Humanoid Robots, (Humanoids 2011), IEEE, 2011, pp. 364–370. http://dx.doi.org/10.1109/Humanoids.2011.6100862.

[14] A.J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal, Dynamical movement primitives: Learning attractor models for motor behaviors, Neural Comput. 25 (2) (2013) 328–373. http://dx.doi.org/10.1162/NECO_a_00393.

[15] B. Nemec, M. Tamošiūnaite, F. Wörgötter, A. Ude, Task adaptation through exploration and action sequencing, in: IEEE-RAS International Conference on Humanoid Robots, (Humanoids 2009), IEEE, 2009, pp. 610–616. http://dx.doi.org/10.1109/ICHR.2009.5379568.

[16] J. Ernesti, L. Righetti, M. Do, T. Asfour, S. Schaal, Encoding of periodic and their transient motions by a single dynamic movement primitive, in: IEEE-RAS International Conference on Humanoid Robots, (Humanoids 2012), IEEE, 2012, pp. 57–64. http://dx.doi.org/10.1109/HUMANOIDS.2012.6651499.

[17] E. Rückert, A. d'Avella, Learned parametrized dynamic movement primitives with shared synergies for controlling robotic and musculoskeletal systems, Front. Comput. Neurosci. 7 (138) (2013) http://dx.doi.org/10.3389/fncom.2013.00138.

[18] D. Velychko, D. Endres, N. Taubert, M.A. Giese, Coupling Gaussian process dynamical models with product-of-experts kernels, in: 24th International Conference on Artificial Neural Networks and Machine Learning (ICANN), in: Lecture Notes in Computer Science, vol. 8681, Springer, Heidelberg, 2014, pp. 603–610. http://dx.doi.org/10.1007/978-3-319-11179-7_76.

[19] J. Denk, G. Schmidt, Synthesis of a walking primitive database for a humanoid robot using optimal control techniques, in: IEEE-RAS International Conference on Humanoid Robots (Humanoids 2001), pp. 319–326, 2001.

[20] J. Denk, G. Schmidt, Walking primitive databases for perception-based guidance control of biped robots, Eur. J. Control 13 (2–3) (2007) 171–188.

[21] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa, Biped walking pattern generation by using preview control of zero-moment point, in: IEEE International Conference on Robotics and Automation, (ICRA 2003), vol. 2, IEEE, 2003, pp. 1620–1626. http://dx.doi.org/10.1109/ROBOT.2003.1241826.

[22] G. Bessonnet, S. Chessé, P. Sardain, Optimal gait synthesis of a seven-link planar biped, Int. J. Robot. Res. 23 (10–11) (2004) 1059–1073. http://dx.doi.org/10.1177/0278364904047393.

[23] K.H. Koch, Using model-based optimal control for conceptional motion generation for the humannoid robot HRP-2 14 and design investigations for exo-skeletons (Ph.D. thesis), Heidelberg University, 2015.

[24] M. Hardt, O. von Stryk, D. Wollherr, M. Buss, Development and control of autonomous, biped locomotion using efficient modeling, simulation, and optimization techniques, in: IEEE International Conference on Robotics and Automation, (ICRA 2003), vol. 1, IEEE, 2003, pp. 1356–1361. http://dx.doi.org/10.1109/ROBOT.2003.1241780.

[25] Y. Xiang, J.S. Arora, K. Abdel-Malek, Physics-based modeling and simulation of human walking: a review of optimization-based and other approaches, Med. Bioeng. Appl. 42 (2010) 1–23.

[26] D. Clever, K. Mombaur, A new template model for optimization studies of human walking on different terrains, in: 2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids), IEEE, 2014, pp. 500–505.

[27] D. Clever, K. Mombaur, An inverse optimal control approach for the transfer of human walking motions in constrained environment to humanoid robots, in: Proceedings of Robotics: Science and Systems, Ann Arbor, Michigan, 2016. http://dx.doi.org/10.15607/RSS.2016.XII.005.

[28] L. Roussel, C. Canudas-De-Wit, A. Goswami, Generation of energy optimal complete gait cycles for biped robots, in: IEEE International Conference on Robotics and Automation, (ICRA 1998), vol. 3, IEEE, 1998, pp. 2036–2041. http://dx.doi.org/10.1109/ROBOT.1998.680615.

[29] G. Schultz, K. Mombaur, Modeling and optimal control of human-like running, IEEE/ASME Trans. Mechatronics 15 (2010) 783–792. http://dx.doi.org/10.1109/TMECH.2009.2035112.

[30] T. Erez, E. Todorov, Trajectory optimization for domains with contacts using inverse dynamics, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2012), IEEE, 2012, pp. 4914–4919. http://dx.doi.org/10.1109/IROS.2012.6386181.

[31] O.E. Ramos, L. Saab, S. Hak, N. Mansard, Dynamic motion capture and edition using a stack of tasks, in: IEEE-RAS International Conference on Humanoid Robots, (Humanoids 2011), IEEE, 2011, pp. 224–230. http://dx.doi.org/10.1109/Humanoids.2011.6100829.

[32] B. Lim, J. Lee, J. Kim, M. Lee, H. Kwak, S. Kwon, H. Lee, W. Kwon, K. Roh, Optimal gait primitives for dynamic bipedal locomotion, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2012), IEEE, 2012, pp. 4013–4018. http://dx.doi.org/10.1109/IROS.2012.6385753.

[33] R. Featherstone, Rigid Body Dynamics Algorithms, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[34] P.-B. Wieber, F. Billet, L. Boissieux, R. Pissard-Gibollet, The HuMAnS toolbox, a homogenous framework for motion capture, analysis and simulation, in: Internal Symposium on the 3D Analysis of Human Movement, 2006.

[35] K.H. Koch, K. Mombaur, O. Stasse, P. Souères, Optimization based exploitation of the ankle elasticity of HRP-2 for overstepping large obstacles, in: IEEE-RAS International Conference on Humanoid Robots, (Humanoids 2014), IEEE, 2014, pp. 733–740. http://dx.doi.org/10.1109/HUMANOIDS.2014.7041444.

[36] H. Bock, K. Plitt, A multiple shooting algorithm for direct solution of optimal control problems, in: 9th IFAC World Congress Budapest, Pergamon Press, Oxford, 1984, pp. 243–247.

[37] D.B. Leineweber, The theory of MUSCOD in a nutshell (Master's thesis), IWR University of Heidelberg, 1995.

[38] D.B. Leineweber, I. Bauer, H. Bock, J. Schlöder, An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization—Part I: theoretical aspects, in: Computers & Chemical Engineering, Vol. 27, Elsevier, 2003, pp. 157–166.

[39] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawaa, T. Kawasaki, M. Hirata, K. Akachi, T. Isozumi, Humanoid robot HRP-2, in: International Conference on Robotics and Automation, (ICRA 2004), vol. 2, IEEE, 2004, pp. 1083–1090. http://dx.doi.org/10.1109/ROBOT.2004.1307969.

[40] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, The MIT Press, 2006.

[41] C.M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, New York, 2007.

[42] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, 2006.

[43] K.B. Petersen, M.S. Pedersen, The matrix cookbook, version 20121115 (nov 2012). URL http://www2.imm.dtu.dk/pubdb/p.php?3274.

[44] R.H. Byrd, P. Lu, J. Nocedal, C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM J. Sci. Comput. 16 (5) (1995) 1190–1208. http://dx.doi.org/10.1137/0916069.

[45] T.E. Oliphant, Python for scientific computing, Comput. Sci. Eng. 9 (3) (2007) 10–20. http://dx.doi.org/10.1109/MCSE.2007.58.

[46] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, Y. Bengio, Theano: new features and speed improvements, Deep Learning and Unsupervised Feature Learning NIPS Workshop, 2012.

[47] A. Mukovskiy, W.M. Land, T. Schack, M.A. Giese, Modeling of predictive human movement coordination patterns for applications in computer graphics, J. WSCG 23 (2) (2015) 139–146.

[48] D. Velychko, D. Endres, The variational coupled gaussian process dynamical model, in: NIPS Workshop on Advances in Approximate Bayesian Inference, 2015. URL http://approximateinference.org/accepted/.

Debora Clever is a postdoctoral researcher at the Interdisciplinary Center of Scientific Computing (IWR) at Heidelberg University in the research group "Optimization in Robotics and Biomechanics" and part of the consortium of the EU-project KoroiBot. She has studied Techno-Mathematics with major on Numerical Analysis, Computer Science and Cybernetics at the Technische Universitaet Darmstadt. There, she received her bachelor's degree in 2005, her diploma degree in 2006 and her Ph.D. in Applied Mathematics with focus on optimal control with partial differential equations in 2013. Her research focus is on inverse optimal control for human gait analysis and on optimal control for humanoid gait generation.

Monika Harant is currently a Ph.D. student in the research group "Optimization in Robotics and Biomechanics" of Prof. K. Mombaur at Interdisciplinary Center of Scientific Computing (IWR) Heidelberg University after receiving her diploma in Mathematics with major in Numerical Analysis and Optimization at Heidelberg University in 2015.

Her research interests include modeling of multibody systems and biomechanical systems, sensorimotor control and learning, optimization-based motion generation for humanoid robots and humanoid push recovery.

K. Henning Koch is presently a research engineer at CC/EYN2 (Chassis System Control Engineering, Networked Systems) of Robert Bosch GmbH in Abstatt, GERMANY. Previously he was a post-doctoral research fellow in the group ORB (Optimization in Robotics and Biomechanics) of Prof. K. Mombaur at IWR (Interdisciplinary Center of Scientific Computing) Heidelberg University, GERMANY.

He holds an interdisciplinary Ph.D. degree on Applied Mathematics and Robotics from Heidelberg University (2015) and a franco-german double-diploma degree in Mechanical Engineering from KIT (Karlsruhe Institute of Technology, Karlsruhe, GERMANY) and the ENSAM ParisTech (Metz/Paris, FRANCE) (2010). The research interests of his Ph.D. work were mechanical modeling of multibody systems for optimization-based motion generation for humanoid robots and conceptional engineering design for lower-limb exo-skeletons. His current research interests are in behavioral planning for engineering of future driver assistance systems.

Katja Mombaur is a professor at the Interdisciplinary Center for Scientific Computing (IWR) at the University of Heidelberg in Germany, where she heads the Optimization in Robotics & Biomechanics (ORB) group and the Robotics Lab. She holds a diploma degree in Aerospace Engineering from the University of Stuttgart (1995) and a Ph.D. degree in Mathematics from the University of Heidelberg (2001). In 2002, she was a Postdoctoral Researcher in the Robotics Lab at Seoul National University, South Korea. From 2008 to 2010 she worked as a visiting researcher at LAAS-CNRS in Toulouse, France.

Katja Mombaur is founding chair of the IEEE RAS technical committee ''Model-based optimization for robotics''. She is coordinating the EU Project KoroiBot, partner in the EU projects MOBOT and SPEXOR, as well as PI and Executive Committee Member of the Graduate School HGS MathComp at IWR. She was a fellow of Marsiliuskolleg at the University of Heidelberg in 2015/16. Her research interests include multibody modeling and optimal control of dynamic motions of humans and humanoid robots with a particular focus on walking and running motions, stability optimization of motions, optimization of design and control of wearable robots as well as inverse optimal control and its application to human movement studies.

Dominik Endres studied Physics at the University of Würzburg, graduating in 1998 with a thesis on Computational Neurophysics. Afterwards he commenced part-time studies for Ph.D. in Computational Neuroscience at the School of Psychology, University of St. Andrews, UK, advised by Dr. Peter Földiak, graduating in 2006. During that time, he also co-founded and ran a small IT-business. After Post-Doc positions in St. Andrews and Tübingen (with Prof. M.A. Giese), where he worked on computational vision and motor control, he accepted a Junior Professorship in Theoretical Neuroscience at the University of Marburg in 2014, where he has been since.