Trajectory generation for multi-contact momentum-control

Alexander Herzog[1], Nicholas Rotella[2], Stefan Schaal[1,2], Ludovic Righetti[1]

Abstract—Simplified models of the dynamics, such as the linear inverted pendulum model (LIPM) have proven to perform well for biped walking on flat ground. However, for more complex tasks the assumptions of these models can become limiting. For example, the LIPM does not allow to control contact forces independently, is limited to co-planar contacts and assumes that the angular momentum is zero. In this paper, we propose to use the full momentum equations of a humanoid robot in a trajectory optimization framework to plan its center of mass, linear and angular momentum trajectories. The model also allows to plan desired contact forces for each end-effector in arbitrary contact locations. We extend our previous results on LQR design for momentum control by computing the (linearized) optimal momentum feedback law in a receding horizon fashion. The resulting desired momentum and the associated feedback law is then used in a hierarchical whole body control approach. Simulation experiments show that the approach is computationally fast and is able to generate plans for locomotion on complex terrains while demonstrating good tracking performance for the full humanoid control.

I. INTRODUCTION

Humanoid robots locomoting and performing manipulation tasks on uneven ground are required to actively apply forces on objects and the floor in order to achieve their task successfully. A direct effect of contact forces is a change of momentum in the robot, which on the one hand is necessary to move the center of mass, but on the other hand restricts the type of limb motion that the robot can perform. The non-linear nature of the angular momentum dynamics makes preview based control computationally hard in general and even with an admissible angular momentum trajectory open control parameters like the joint motion and feedback control remain problematic for use in a whole body control framework.

Successful applications of simplified momentum models have been shown on robots walking quite robustly over flat ground. A common approach is to use the linear inverted pendulum model (LIPM), which has been exploited for preview control since it was introduced by Kajita et. al [1]. In [2] a model predictive control (MPC) approach is formulated that finds foot steps on a flat ground together with a compatible CoM location on a horizontal plane. CoM profiles can then be realized on the full body together with other limb motion (e.g. swinging leg) controllers [3], [4] [5], [6]. The assumption of a horizontal CoM motion and flat ground can be relaxed to a pre-designed CoM height profile [7], [8]. Although, less restrictive than the original LIPM, these approaches are either built for point feet or leave parts of the dynamics uncontrolled. Depending on the terrain, pre-defining the CoM along a fixed direction may result in suboptimal or infeasible reaction forces. Approaches that leave the regime of linear models have been proposed, e.g. for long jumps [9] leading to more complex task behavior, but require task specific models that e.g. take into account the swing leg dynamics. Going even further, we have seen work that optimizes over the whole joint trajectories and the full momentum together [10]. The impressive near-to physical motions, however, come at a high computational cost. Time-local controllers, have shown that balancing performance of robots is improved, when overall angular momentum is damped out directly [11], [12], however it has remained unclear how angular momentum profiles should be chosen in non-static configurations.

In this paper we consider the full momentum dynamics of the robot for generation of CoM and momentum trajectories and the according reaction force profiles at each contact. We formulate the problem as continuous time optimal control problem in a sequential form. A mode schedule is predefined together with end effector trajectories. A desired angular momentum trajectory, which is required for the limb motion, is generated from a simple inverse kinematics forward integration and realized with admissible contact forces in a least squares optimal sense. Since the kinematics information is used before the optimization over the dynamics, our optimization procedure is relatively fast (e.g. compared to [10]). In our previous work [12] we have proposed a way to compute control gains for the momentum task using a LQR design approach. We showed that it was able to significantly improve performance for balancing tasks on a real humanoid. However, this approach was only used for stabilization. In this paper, we extend the approach to receding horizon tracking control. We use the planned trajectories to generate feedback gains using a LQR design, where we linearize the non-linear momentum dynamics around the optimized trajectory. In a simulated stepping task our humanoid traverses a terrain with changing height and tilted stepping stones successfully. A whole-body controller finds joint torques that realize feedback loops on momentum as well as the swing leg motion and respect additional balancing, hardware constraints and generates physically realizable contact forces. A good tracking of overall momentum is achieved using our receding horizon LQR design.

[1] Autonomous Motion Department, Max Planck Institute for Intelligent Systems, Tübingen, Germany. `first.lastname@tuebingen.mpg.de`
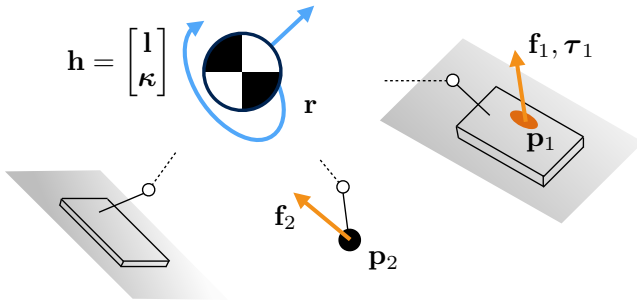[2] CLMC Lab, University of Southern California, Los Angeles, USA.

Fig. 1. A sketch of the model used through out this paper. It shows the center of mass (CoM) of the system and contacts with the environment. Flat contacts that consist of a force, center of pressure and normal torque can be modeled (e.g. a hand touching a wall) as well as point contacts. There is no restriction on the pose of contacts nor the CoM. Momenta generated around the CoM are not necessarily zero.

This paper is organized as follows. In Sec II we formulate an optimal control problem to obtain momentum trajectories of the robot. Resulting trajectories are controlled with a LQR feedback design as discussed in Sec III. It is incorporated into a whole-body control approach as explained in Sec IV. In Sec V we demonstrate our control framework on a simulated stepping task on rough terrain. We discuss our results in Sec VI and finish with a conclusion in Sec VII.

II. MOMENTUM TRAJECTORY OPTIMIZATION

In this section we describe how the robot momentum is planned together with admissible contact forces. We will first discuss the reduced dynamics model which will then be used to phrase an optimal control problem to generate CoM and momentum profiles. At the end of the section we describe how a desired angular momentum is chosen for optimization.

A. Dynamic Model

We reduce the state of a robot to its center of mass (CoM) $\mathbf{r}$ and overall momentum

$$\mathbf{h} = \begin{bmatrix} \mathbf{l} \\ \boldsymbol{\kappa} \end{bmatrix}, \tag{1}$$

with linear $\mathbf{l}$ and angular $\boldsymbol{\kappa}$ momenta. The state of the system is controlled by forces $\mathbf{f}_i$ and torques $\boldsymbol{\tau}_i$ acting at points $\mathbf{p}_i$ on the mechanical structure. The momentum dynamics as illustrated in Fig 1, can be written

$$M\dot{\mathbf{r}} = \mathbf{l} \tag{2}$$
$$\dot{\mathbf{l}} = M\mathbf{g} + \sum \mathbf{f}_i \tag{3}$$
$$\dot{\boldsymbol{\kappa}} = \sum \boldsymbol{\tau}_i + \sum (\mathbf{p}_i - \mathbf{r}) \times \mathbf{f}_i, \tag{4}$$

where M is the mass of the robot. The contact points $\mathbf{p}_i$ can for instance be point foot locations that are touching the ground or they can represent the position of a handle that the robot holds on to. Flat feet can be modeled by several contact points on the foot surface. Equivalently, we can represent the effective force at the center of pressure (CoP). Depending on the type of contact, it will be necessary to express constraints on the forces. In the case of point feet, no torques $\boldsymbol{\tau}_i$ can

be generated. Torques that are generated at the CoP of a flat foot are required to be normal to the foot surface. Further, if we assume stationary contact points, we need to restrict the wrench to remain in a friction cone.

B. Optimal control formulation

In the following we use the momentum dynamics of the robot to plan for admissible contact forces that generate desired (linear and angular) momentum trajectories. Our goal is to compute force trajectories $\mathbf{f}_i(t), \boldsymbol{\tau}_i(t), \mathbf{p}_i(t)$ that satisfy the momentum dynamics in Eqs (2-4) and contact constraints at all time. Also, they are required to steer the momentum through out desired states over a time horizon T. Given the initial state of the robot CoM and momentum $\mathbf{r}(0), \mathbf{l}(0), \boldsymbol{\kappa}(0)$, we can integrate Eqs (2-4) to obtain

$$\dot{\mathbf{l}}(t) = \int_0^t (\sum \mathbf{f}_i(\delta) + M\mathbf{g}) d\delta, \tag{5}$$
$$\mathbf{l}(t) = \mathbf{l}_0 + \int_0^t \dot{\mathbf{l}}(\delta) d\delta, \tag{6}$$
$$\mathbf{r}(t) = \mathbf{r}_0 + \frac{1}{M} \int_0^t \mathbf{l}(\delta) d\delta, \tag{7}$$
$$\dot{\boldsymbol{\kappa}}(t) = \int_0^t (\sum \boldsymbol{\tau}_i(\delta) + \tag{8}$$
$$\sum (\mathbf{p}_i(\delta) - \mathbf{r}(\delta)) \times \mathbf{f}_i(\delta)) d\delta,$$
$$\boldsymbol{\kappa}(t) = \boldsymbol{\kappa}_0 + \int_0^t \dot{\boldsymbol{\kappa}}(\delta) d\delta \tag{9}$$

Note that the states are expressed as (nested) functions of the reaction forces. They are constructed by integrating, summing and applying the cross-product on $\mathbf{f}_i(t), \boldsymbol{\tau}_i(t), \mathbf{p}_i(t)$. Given a naive idea of what the desired CoM $\mathbf{r}_{des}(t)$ and momentum $\mathbf{h}_{des}(t)$ should be (e.g. coming from an initial kinematic plan), we want to find contact forces that minimize the error

$$\begin{aligned} J = \sum_{t_0}^T (& \|\dot{\mathbf{l}}(t_i)\|_{W_1}^2 + \|\mathbf{l}(t_i) - \mathbf{l}_{des}(t_i)\|_{W_2}^2 + \\ & \|\mathbf{r}(t_i) - \mathbf{r}_{des}(t_i)\|_{W_3}^2 + \|\dot{\boldsymbol{\kappa}}(t_i)\|_{W_4}^2 + \\ & \|\boldsymbol{\kappa}(t_i) - \boldsymbol{\kappa}_{des}(t_i)\|_{W_5}^2 \quad), \end{aligned} \tag{10}$$

where we compute the errors at $t_i \in [0; T]$ and W_i are diagonal weighting matrices that allow trade-offs between the cost terms.

Adaptive end effector location: Through out the discussion in this paper we assume that the end effector remains at a predefined location. However, this assumption can be relaxed without making the optimization problem harder. For that we simply substitute $\mathbf{p}_i = \bar{\mathbf{p}}_i + \tilde{\mathbf{p}}_i$, where $\bar{\mathbf{p}}_i$ is the foot sole location that remains stationary through out a contact phase and $\tilde{\mathbf{p}}_i$ is the (time-varying) center of pressure inside of the foot sole. In this reformulation both, $\bar{\mathbf{p}}_i$ and $\tilde{\mathbf{p}}_i$, will be optimization variables and are chosen automatically. The support planes still have to be decided before-hand, e.g. using a dedicated acyclic contact planer [13], [14].

C. Optimization procedure

We will now formulate the described problem into an optimization problem. First, the reaction forces are formulated as weighted basis functions, more specifically as polynomials of the form

$$\mathbf{f}(t; \mathbf{w}) = \alpha(t) \sum_{k=0}^{N-1} w_k t^k = \Phi^T(t) \mathbf{w}, \qquad (11)$$

$$\alpha(t) = \begin{cases} 1 & \text{if in contact at } t \\ 0 & \text{else} \end{cases}, \qquad (12)$$

with weights w_i and basis functions t^k summarized in vectors $\Phi(t), \mathbf{w}$. We define mode-scheduling $\alpha(t)$, which specifies contact activation and deactivation and is set beforehand, e.g. it could be obtained by a higher level planner. Polynomials have the advantage of generating smooth force trajectories by construction. The forces can then be written as

$$f_i^j(t; \mathbf{w}_i^j) = \Phi^T(t) \mathbf{w}_i^j, \ j \in x, y, z \qquad (13)$$
$$\tau_i(t; \mathbf{v}_i) = \mathbf{n}_i \Phi^T(t) \mathbf{v}_i \qquad (14)$$
$$p_i^j(t; \mathbf{u}_i^j) = \Phi^T(t) \mathbf{u}_i^j, \ j \in x, y \qquad (15)$$

where the subscript identifies the end effector, the superscript represents the coordinates and $\mathbf{n}$ is the normal vector of the foot sole. Contact forces, torques and CoPs, now become linear functions of polynomial coefficients. Expressing the states in Eqs (5-9) with Eqs (13-15) substituted, gives us the states as functions of time and polynomial coefficients. The operations required to carry out the result boil down to multiplication, summation and integration of polynomials, which can be computed analytically. As a result we phrased our optimal control problem in sequential form, where our controls evaluate directly to states and the dynamics equations (cf. Eqs (2-3)) are implicitly incorporated. This allows us to phrase our optimization problem

$$\min_{\mathbf{x}} \ J(\mathbf{x}) \qquad (16)$$
$$\text{s.t.} \quad |p^j(t_i; \mathbf{x})| \le \hat{p}, \ j = x, y \qquad (17)$$
$$|\tau(t_i; \mathbf{x})| \le \hat{\tau}, \qquad (18)$$
$$0 \le -|f^j(t_i; \mathbf{x})| + \mu f^z(t_i; \mathbf{x}) \le \hat{f}_z, \ j = x, y \quad (19)$$
$$t_i = t_0 \dots T$$

where $\mathbf{x}$ is a concatenation of variables $\mathbf{w}_i^j, \mathbf{v}_i, \mathbf{u}_i^j$. We try to find polynomial coefficients $\mathbf{x}$ that minimize the error cost J, while satisfying friction and support bound constraints on forces. Eqs (17-18) bound the CoPs and torques. In Eq (19) we impose a friction cone constraint approximated as pyramid and upper bound it by a sufficiently large value $\hat{f}_z$ to keep the polynomials from penetrating the lower bound and escaping above. In our constraints in Eqs (17-19), we wish to have bounds that hold at all $t \in [0, T]$, in practice however, we express them at a finite number of time steps. Given the

limited flexibility of polynomials, we get minor penetration of those constraints in intervals $(t_i; t_{i+1})$. Note that all our constraints are linear, whereas the objective function has quadratic terms as well as higher order (non-convex) terms due to the costs on angular momentum. Here we have the benefit that all terms in the momentum dynamics can be included in the cost and none are left uncontrolled.

Receding Horizon: Given that the problem in Eq (16) is non-convex, we cannot expect to find a global optimum in general, but need to find a local optimum starting from an initial guess $\mathbf{x}_0$. In our approach we use a receding horizon technique. We start out with solving our problem for a horizon $\tilde{T} < T$ and obtain an optimal solution $\mathbf{x}_{[0, \tilde{T}]}^*$. Typically we start our desired motion such that it is easy to solve in the interval $[0, \tilde{T}]$, i.e. quasi static solutions are already optimal. Then, we formulate our problem for the interval $[\Delta, \tilde{T} + \Delta]$, where we initialize our optimizer with the previous solution. This has the advantage that an initial solution can be bootstrapped from an initial easier to solve configuration and then moved over the horizon to the difficult parts of the motion. Further as we push our implementation to more speed, we seek to use it in a receding horizon control setting.

Desired Angular Momentum: Most simplified momentum models assume that angular momentum is desired to remain at zero as much as possible. This however ignores the fact that momentum might be required to move the robot limbs, e.g. in a stepping task we need to swing a leg, which in turn generates a momentum around the hip. Thus, it is not trivial to decide what the desired angular momentum should look like. In order to overcome this issue, we integrate forward our desired swing leg trajectories using inverse kinematics. From the resulting joint motion we then compute $\mathbf{r}_{des}(t), \dot{\kappa}_{des}(t)$ (explained e.g. in [15]). Although, dynamically not feasible, this method generate angular momentum profiles required to perform the task imposed motion, e.g. swinging a leg. We iterate between kinematic forward integration of an optimized $\mathbf{r}(t)$ and optimizing a desired $\boldsymbol{\kappa}$ with dynamic constraints (by solving the problem in Eqs (16-19)). In our experiments two passes already lead to convergence. With this iteration we not only generate admissible force profiles that obey the momentum dynamics equations, but also we can bootstrap angular momentum trajectories which are not obvious to design otherwise. Note that neither CoM nor momentum and neither components of the two are predefined in advance. Instead an initial guess is given and the final trajectories are found automatically.

III. MOMENTUM LQR

In the previous section we described how reference trajectories in accordance to the momentum dynamics are obtained. In order to track those trajectories on the full robot, we propose a feedback law using a LQR design, which has shown superior performance, compared to a naive PD gain approach in experiments on the real robot [12]. From our trajectory optimizer we obtain admissible states and controls

$$\mathbf{y}^* = \begin{bmatrix} \mathbf{r} \\ \mathbf{l} \\ \boldsymbol{\kappa} \end{bmatrix}, \boldsymbol{\lambda}^* = \begin{bmatrix} \vdots \\ \mathbf{f}_i \\ \boldsymbol{\tau}_i \\ \vdots \end{bmatrix}, \tag{20}$$

$$\dot{\mathbf{y}}^* = f(\mathbf{y}^*, \boldsymbol{\lambda}^*) = \begin{bmatrix} \frac{1}{M}\mathbf{l} \\ M\mathbf{g} + \sum \mathbf{f}_i \\ \sum \boldsymbol{\tau}_i + \sum (\bar{\mathbf{p}}_i - \mathbf{r}) \times \mathbf{f}_{i,} \end{bmatrix} \tag{21}$$

where we transform wrenches $\mathbf{f}_i, \boldsymbol{\tau}_i$ to the stationary poles $\bar{\mathbf{p}}_i$.

The dynamics function in Eq (21) is discretized an linearized around the desired trajectories $\mathbf{x}^*, \boldsymbol{\lambda}^*$. The resulting time varying linear dynamics are then used to formalize a finite horizon LQR problem. This yields a control policy

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}^* + \mathbf{K}_t(\mathbf{x} - \mathbf{x}^*) \tag{22}$$

with time varying feedforward and feedback terms that map errors in states into contact wrenches. Controlling the momentum with this feedback law requires 6 DoF for wrenches at each contact. Since our whole-body controller incorporates additional control objectives and force constraints, we compute directly the momentum rate that the wrenches generate

$$\dot{\mathbf{h}}_{ref} = \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\ [\bar{\mathbf{p}}_i - \mathbf{r}]_\times & \mathbf{I}_{3\times 3} & \cdots \end{bmatrix} \mathbf{K}_t(\mathbf{x} - \mathbf{x}^*) + \dot{\mathbf{h}}^*, \tag{23}$$

where $[\,\boldsymbol{\cdot}\,]_\times$ turns a cross-product into a matrix multiplication. The resolution of momentum rate to contact wrenches is then left to our whole-body controller as described in the next section. In this LQR design a quadratic performance cost is set once and then optimal gains are computed at each time step. As we discussed in our previous work, in order to achieve compatible results with diagonal PD gain matrices, we had to design different sets of gains, whereas the LQR design requires one performance cost and generates suitable gains at different contact configurations. Different to our previous work we linearize around desired trajectories, whereas in our robot experiments we used only two key configurations, which may become limiting in more versatile tasks such as walking over a rough terrain.

IV. WHOLE-BODY CONTROL

The trajectories generated with the model in Eqs (2-4) define the CoM, momentum and end effector forces of our humanoid. In order to track these trajectories on the full robot, we need to generate joint torques accordingly and at the same time control the limb motion and guarantee that other constraints are obeyed, e.g. joint limits. For this time-local control problem, we use inverse dynamics in QP Cascades, which we applied successfully on the real robot before [6]. It allows us to phrase feedback controllers and constraints as functions of joint accelerations $\ddot{\mathbf{q}}$, external generalized forces $\boldsymbol{\lambda}$ and joint torques $\boldsymbol{\tau}$. For instance, we can express a cartesian controller on the swing foot as affine

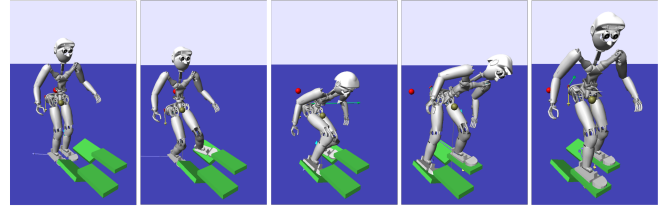Fig. 2. The humanoid robot traversing a terrain with stepping stones of different hight and orientation.

function of joint accelerations or a momentum controller as a function of reaction forces. The latter e.g. would be

$$\begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\ [\bar{\mathbf{p}}_i - \mathbf{r}]_\times & \mathbf{I}_{3\times 3} & \cdots \end{bmatrix} \boldsymbol{\lambda} + \begin{bmatrix} M\mathbf{g} \\ \mathbf{0} \end{bmatrix} = \dot{\mathbf{h}}_{ref}. \tag{24}$$

A reference momentum rate $\dot{\mathbf{h}}_{ref}$ can be chosen from a LQR design as demonstrated in our previous work [12] and extended in Sec III. Given a set of controllers and tasks, we find torques that satisfy the dynamics equations of the full robot and at the same time generate the desired task feedback as good as possible. Tasks, however, may conflict, e.g. moving the CoM may require moving the swing leg and vice versa. In these cases, QP cascade allow two types of trade-offs, where we can either weigh tasks against each other or we can prioritize them strictly. Especially, when it comes to trade-offs between constraints, tasks of interest and redundancy resolution, prioritization can facilitate the control design.

V. SIMULATION RESULTS

This section describes our simulation results of the proposed control framework. We use a model of our Sarcos humanoid in the SL simulation environment. Contacts are simulated with a penalty method and stiff springs. All experiments are performed on a 2.7 GHz intel i7 processor with 16gb ram. A task is generated where the robot is to walk on stepping stones that increase in height from one step to the other as visualized in Fig 2 and shown in the attached video[1]. The z-axis of the inertial frame points up and the robot walks along the y-axis to the front. Two out of the four steps are tilted by 25°. Both, the change in CoM height as well as the tilted support break the assumptions made in LIPM models and require to consider the two contact wrenches in separate. We generate swing leg trajectories using cubic splines on the pose of the foot. The humanoid starts out in double support at rest. After the first 3 seconds the left foot breaks contact and moves on a tilted stone located to the front left-hand side of the robot at a lifted height (as shown in Fig 2). It follows a contact switch after each second changing from single- to double support or vice versa. The second step is again a stone tilted inwards to the robot and located to the front right-hand side. Finally, the robot takes two steps onto

[1]An extended version of the video is available on `www-amd.is.tuebingen.mpg.de/~herzog/15_07-Humanoids.mp4`

Rank	Nr. of eq/ineq constraints	Constraint/Task
1	6 eq	Newton Euler Equation
2	2 × 6 eq	Contact constraints
	2 × 4 ineq	Center of Presure
	2 × 4 ineq	Friction cone
	2 × 14 ineq	joint acceleration limits
3	6 eq	LQR momentum control
	6 eq	Cartesian swing foot control
	14 + 6 eq	PD control on posture
4	2 × 6 eq	Contact forces control
5	3 eq	Base link orientation

TABLE I

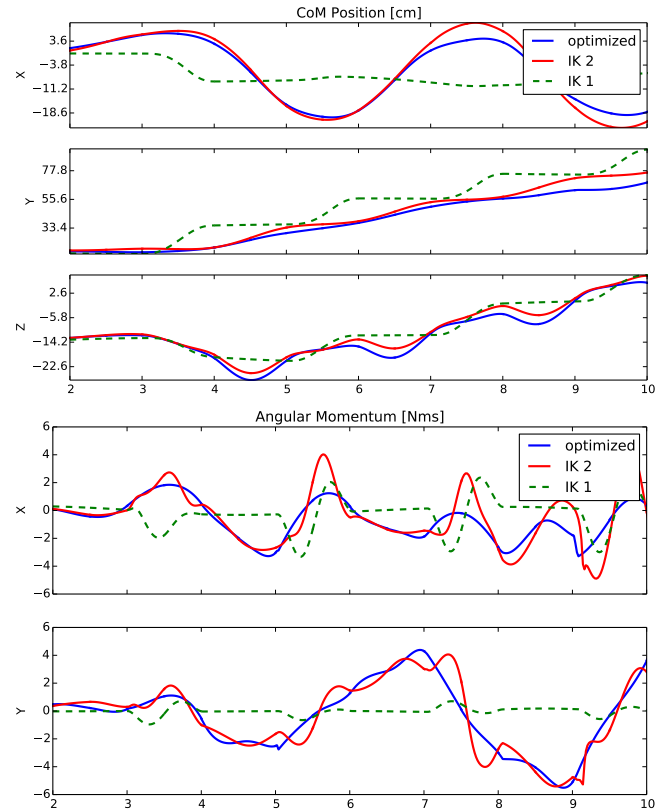HIERARCHY FOR TASKS IN THE STEPPING EXPERIMENT

Fig. 3. Plans for the CoM and horizontal angular momentum. The resulting lateral CoM (top plot) was adapted quite significantly over the planning process to make it physically compatible with the reaction forces. Angular momenta are found that allow for stepping motions required to traverse the terrain.

a horizontal plateau located slightly below knee height.

The planner is initialized with a naive idea of the robot motion, where we simply keep the base at a certain height above the feet. After integrating the desired swing foot positions together with the base height using inverse kinematics (as described in Sec II-C), we obtain resulting $\mathbf{r}_{des}(t), \boldsymbol{\kappa}_{des}(t)$ trajectories that take into account the angular momentum required to swing the legs from one stepping stone to the other. The inverse kinematics solution, which is physically not consistent, is then adjusted in the trajectory optimization step to be admissible with constrained forces and CoPs. None of the force constraints were notably violated. As can be seen in Fig 3 the inverse kinematics generated CoM trajectory is modified significantly in the lateral direction and as a consequence the angular momentum in y-direction as well. After a second iteration of inverse kinematics integration and trajectory optimization the results have sufficiently converged and we stop. With polynomials of order 3 the control problem had a total of 72 variables. The planning process took 4 min and converged after only two iterations. This allows us to generate a complex motion rather quickly compared to motion planners that are based on more extensive models of the robot. The numerical optimization problem in Eq 16 is solved with SNOPT, a Sequential Quadratic Programming method.

Next, we construct a hierarchy of feedback controllers and constraints in order to realize the momentum profile on the full humanoid. At the highest priority we express the physical model of the full robot to obtain physically consistent torques. It follow force and joint limits together with contact constraints. In the third priority we control the swing leg motion and the momentum and add a posture PD control with a relatively low weight. In the priorities below we regularize end effector forces and stabilize the base orientation. A summary of the task setup is given in Tab I. The momentum is controlled with feedback gains designed as described in Sec III, where our performance cost is fixed for the whole run. We penalize state errors by 10, forces by 0.1 and torques by 0.5. Gains are generated over a 2sec horizon with a granularity of 200 time steps. A sequence of gains is recomputed every 10ms.

The robot was able to traverse the terrain successfully. The momentum trajectories, which were both, dynamically consistent as well as compatible with the robots limb motions, could be tracked well as shown in Figs 6 5. Torques were generated that are in the bounds of the physical robot. In the beginning of the task the planned CoM height forces the knees to stretch, which prevents the robot from using its knees to lift the CoM further, but instead it lifts the arms up rapidly. This can be avoided by adding a box constraint on the CoM in the optimization problem in order to consider kinematic limitations.

The LQR gain matrices have non-zero off-diagonal values as expected (cf. Fig 4). For instance we can see that angular momentum is generated in order to correct for errors in CoM and linear Momentum, which would not be possible with diagonal PD gains. In fact we tried to track the planned motion using diagonal PD gains. We started with values similar to the diagonal of the LQR gain, but we could not find parameters that were stable through out several contact situations as it was the case for the LQR gains. Increasing the terrain difficulty was mainly problematic due to kinematic limits, because our naive swing leg trajectories required to keep the heel on the ground during the whole support phase thus limiting the stepping height. Further, we saw problems with the simulators contact model, which caused sporadic spikes in the force profile when we applied strong forces on the ground.

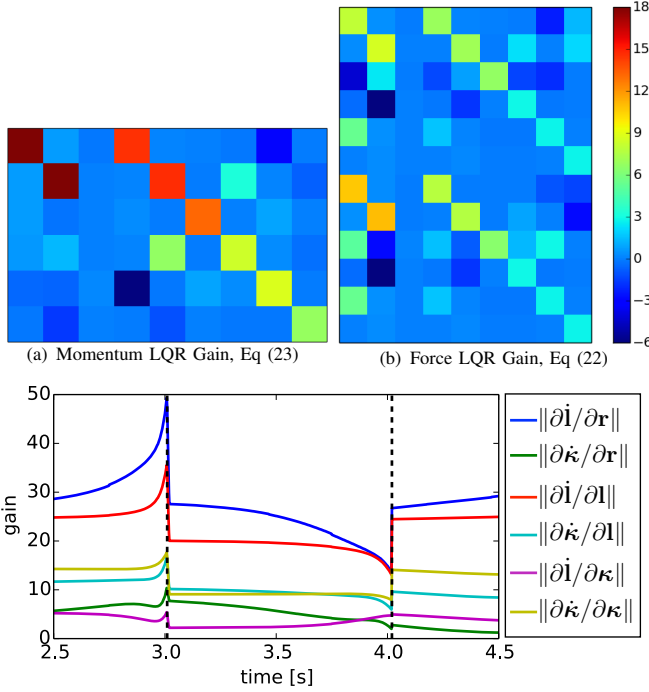(a) Momentum LQR Gain, Eq (23) (b) Force LQR Gain, Eq (22)

Fig. 4. Here we show a heat map visualization of the momentum gain (top left) and the force gain (top right). The top 6 rows in b) correspond to the wrench of the right foot, wheras the bottom rows correspond to the left foot. As we can see the gains contain off-diagonal terms leading to coupling terms between linear and angular momentum. These terms are ignored in a naive diagonal PD gain design. The bottom plot shows the norm of 3x3 sub-blocks of the momentum gain plotted over time. The vertical dashed lines at $t = 3$ and $t = 4$ indicate contact switches. Gain profiles change significantly over time and contact configurations. The gains and momentum dynamics are discontinuous at contact switches. Nevertheless, discontinuities in joint torques were negligible.

Overall, a complex task could be planned quickly including a non trivial angular momentum profile that respected the end effector motion. The proposed feedback control law on the momentum showed good performance when it was embedded in a inverse dynamics task hierarchy.

VI. DISCUSSION

Relation to simplified Momentum Dynamics: The momentum model in Eqs (2-4) is often simplified further in order to obtain linear dynamics, which then leads to computationally more efficient algorithms. However, turning multiplicative terms between variables in Eq (4) into linear terms requires potentially restrictive assumptions. E.g. in the LIPM the CoM height r_z is assumed to be constant, there is one effective $\mathbf{p}_i$, which lies in one horizontal plane, and the force is required to act along $\mathbf{r} - \mathbf{p}$.

Since a constant CoM height may be limiting for tasks that require to move vertically, the authors in [8] allow a predefined (not constant) $r_z(t)$. Substituting this assumption into Eq (4), and assuming constant $\mathbf{p}_i$, turns the horizontal angular momentum into a linear function of forces and torques. Assuming in addition, that the non-linear vertical angular momentum can be neglected, the authors end up again with a linear dynamics. If we restrict our dynamics model
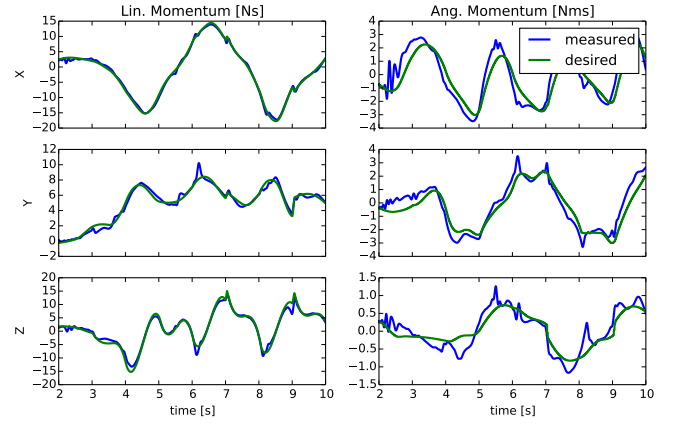

Fig. 5. Linear and angular momentum are tracked well as the robot walks over up the terrain. The oscillations in the angular momentum at t=2 come from rapid arm motions, when the robot was trying to move the CoM up and the knees were stretched. This can be avoided, e.g. by adding box constraints on the CoM in the trajectory optimization step to account for kinematic limits.
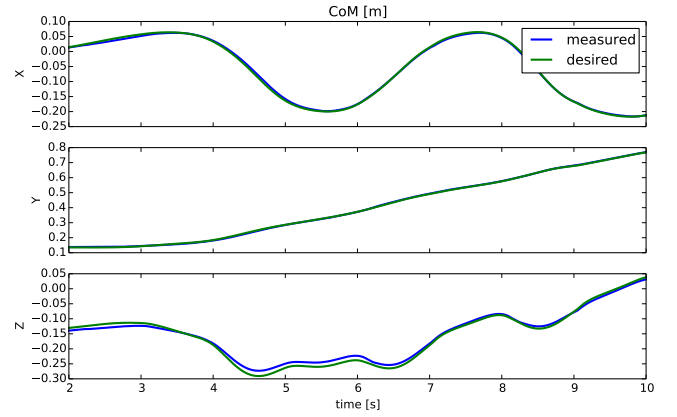

Fig. 6. Measured and desired CoM of the robot, when it was traversing the terrain. As the plots show, good tracking performance can be achieved with the proposed LQR design on the momentum.

further into a LIPM and use discretized dynamics (instead of polynomial trajectories) we can recover the approach of [2]. However, this can be restrictive, if we want to step on various slopes or when a non-zero angular momentum is required, e.g. to consider limb motion.

Computation time: In our current work, we focused on the capabilities of the multi-contact dynamics model to generate dynamic motions on uneven terrain and we showed that these behaviors can be controlled on a full humanoid robot. However, the goal of separating the control process into a predictive control generation on a lower dimensional model and time-local control on the full dynamics, has the potential for a fast implementation in a MPC fashion. In our experiments we saw potential drawbacks in our numerical optimization in Eq (16). Increasing the order of polynomials, leads to slower convergence rates, whereas reducing the polynomial dimensions too far may lead to poor flexibility of the trajectory representation. This may be explained by the discrepancy of basis functions evaluated close to 0 and

close to T leading to s poorly conditioned problem. Piecewise constant trajectories may overcome this problem. Another point for improvement of the numerical procedure is the formulation of the optimal control problem. In the current formulation, the objective function in Eq (10) has terms up to 4th order (squared norm of cross products). Our numerical solver is based on approximations up to second order, which may limit the region, in which approximations are valid. It is possible (but out of the scope of this paper) to rewrite the problem into a (non-convex) Quadratically Constrained Quadratic Program, where the objective function as well as the constraints are quadratic, leading to a better approximation in second order methods and potentially improve convergence. Pushing the implementation towards an online control algorithm is part of our future work.

VII. CONCLUSIONS

We presented an approach to control contact forces and momentum for humanoid robots. CoM and momentum profiles were obtained in an optimal control framework together with admissible contact forces. Feedback gains are generated from a LQR design, which generates time and contact-configuration dependent gains from a single performance cost. The resulting controller is embedded in an inverse dynamics based whole-body controller together with other limb controllers and constraints. We demonstrated the control framework on a simulation of the Sarcos humanoid traversing rough terrain. Physically admissible momentum and force trajectories could be found relatively quickly and were tracked well during the task execution.

In future work, we will implement the discussed speed-up and push the optimal controller to run online.

REFERENCES

[1] S. Kajita, F. Kanehiro, and K. Kaneko, "Biped walking pattern generation by using preview control of zero-moment point," in *ICRA*, 2003.
[2] A. Herdt, N. Perrin, and P.-B. Wieber, "Walking without thinking about it," in *IROS*, 2010, pp. 190–195.
[3] A. Sherikov, D. Dimitrov, and P.-B. Wieber, "Whole body motion controller with long-term balance constraints," *IROS*, 2014.
[4] S. Feng, X. Xinjilefu, W. Huang, and C. Atkeson, "3d walking based on online optimization," in *Humanoids*, 2013.
[5] S. Faraji, S. Pouya, and A. Ijspeert, "Robust and Agile 3D Biped Walking With Steering Capability Using a Footstep Predictive Approach," in *R:SS*, Berkeley, USA, July 2014.
[6] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal, "Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics," in *IROS*, 2014.
[7] J. Englsberger, C. Ott, and A. Albu-Schaffer, "Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.
[8] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, and E. Yoshida, "Model preview control in multi-contact motion-application to a humanoid robot," in *IROS*, 2014, pp. 4030–4035.
[9] P. M. Wensing and D. E. Orin, "Development of high-span running long jumps for humanoids," in *ICRA*, 2014, pp. 222–227.
[10] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body Motion Planning with Simple Dynamics and Full Kinematics," *Humanoids*, 2014.
[11] S.-H. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Autonomous Robots*, vol. 33, pp. 399–414, 2012.
[12] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid," *Autonomous Robots*, (accepted for publication). [Online]. Available: http://arxiv.org/abs/1305.2042v1
[13] T. Bretl, S. Rock, J. Latombe, B. Kennedy, and H. Aghazarian, "Free-climbing with a multi-use robot," in *International Symposium on Experimental Robotics ISER*, 2004.
[14] S. Tonneau, N. Mansard, C. Park, D. Manocha, F. Multon, and J. Pettre, "A reachability-based planner for sequences of acyclic contacts in cluttered environments," in *International Symposium on Robotics Research ISRR*, 2015.
[15] D. E. Orin and A. Goswami, "Centroidal Momentum Matrix of a humanoid robot: Structure and Properties," in *IROS*, 2008.