2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

Linear Contact Modeling and Stochastic Parameter Optimization for LQR-based Whole-Body Push Recovery

Simon Bäuerle, Lukas Kaul and Tamim Asfour

***Abstract*— In this paper we extend the line of research that aims at applying linear optimal control approaches with quadratic cost (LQR) to the inherently non-linear control problem of whole-body balancing for push recovery of humanoid robots. The non-linearity of the system is addressed in the controller design by optimization in the weight-space of the cost function in order to maximize balancing performance. We use stochastic sampling-based, gradient-free optimization over the large design parameter space of the whole-body controller to efficiently cope with the unknown relation between the cost function and the balancing performance. We further investigate three different linear ground contact models and evaluate their influence on the overall controller performance. We demonstrate that parameter optimization and novel ground contact models can be used to design a linear balancing controller that produces human-like whole-body motions in physics simulation-based push recovery experiments, simultaneously considering joint angles, center of mass and angular momentum.**

I. Introduction

Balancing a humanoid robot by means of whole-body motions (*postural balancing*) is a challenging control task due to the high number of Degrees of Freedom (DoF) and the inherent non-linearity of the control problem. State-of-the-art solutions mainly rely on run-time optimizations (e. g. QP-based controllers) that are computationally expensive and therefore require very capable on-board computing hardware. An alternative approach is to use a linear controller which is able to control the non-linear plant over a substantial portion of the state space. Once a linear controller (such as an LQR) is optimized off-line, it has the advantage of very low computational demands at run-time. This in turn ensures fast execution times, real-time capability, and allows for the use of cost efficient control hardware. By encoding the whole-body balance objective as Center of Mass (CoM) position and angular momentum in the LQR cost function, a whole-body balance controller can be realized.

The main research question is how to enable the linear controller to effectively control the movement of a complex system such as a humanoid robot. In this paper we investigate two aspects of the answer to this question: One is to appropriately linearize the system model used for controller synthesis, in particular the ground contact model. We propose two novel models of contact modeling and evaluate them in the context of linear controller design for whole-body balancing. The second aspect is to carefully select the design weights in the LQR cost function. We show that they have a significant effect on controller performance and demonstrate a way to systematically optimize them based on data gathered in dynamics simulation.

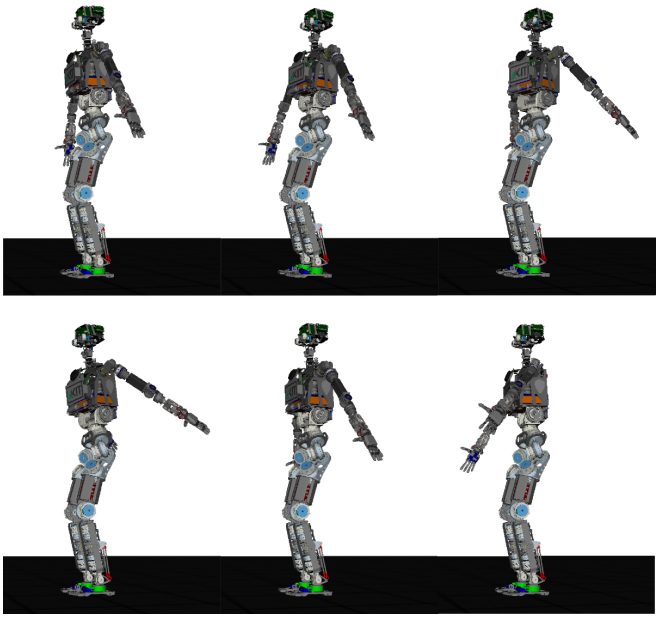


Fig. 1: Simulation of the ARMAR-4 humanoid robot using linear whole-body balance control to recover from a push applied from the front.

A. Related Work

The problem of dynamic balancing in complex humanoid robots can be formulated as constrained optimization problem with quadratic cost (QP): The weighted objectives can be a state-dependent notion of stability (e. g. the center of pressure, center of mass, zero moment point, angular momentum) together with a desired body pose or trajectory. The constraints are typically consistency with the non-linear equations of motion, the maximum admissible friction forces in the ground contact (friction cone) as well as angle and torque limits on joint level. Solving such a QP in discrete time steps produces desired acceleration or torque values for the robot joints that fulfill the objectives while complying with the constraints. This approach received significant attention over the last years and was extensively covered in the respective literature (e. g. [1], [2], [3], [4]). Besides the careful choice and formulation of objectives and constraints, the main downside of QP-based approaches for balancing is the very high computational demand of real-

The authors are with the High Performance Humanoid Technologies Lab, Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany, {lukas.kaul, asfour}@kit.edu

This work has been supported by the German Federal Ministry of Education and Research (BMBF) under the project INOPRO (16SV7665).

time capable QP solvers. Formulating these problems in an efficient-to-solve way as well as efficient solvers themselves are therefore research topics in their own right [5]. More complex frameworks integrating motion planning and QP motion control have also been proposed [6], [7]. Such integrated approaches often internally use an LQR formulation based on simplified CoM dynamics (e. g. point mass model). Simplified abstractions of the robot dynamics such as the Divegent Component of Motion (DCM) [8], the Capture Point (CP) [9] or the Linear Inverted Pendulum (LIPM) [10] are the basis of several methods for balancing. In contrast, the approach we are using in this paper formulates the LQR over the full dynamics model of the robot, taking into account all joints that significantly contribute to balancing (i. e. the legs, the torso and the shoulders).

Another approach to online trajectory optimization is to use model predictive control (MPC) on simplified [11], [12] or more complex models, where promising results have been reported for the use of Differential Dynamic Programming (or variants thereof), albeit often at the cost of very high computational cost for complex humanoid models [13].

Full-body balancing and locomotion control has recently also been addressed with methods relying on data-driven Deep Reinforcement Learning (DRL). Rather than accurately modeling the robot and carefully considering its physical constraints, motion strategies and feedback controllers are iteratively learned and automatically modeled with a deep neural network from data generated in dynamic simulations [14]. This line of research is still in its early stages and needs to show its applicability to actual robotic balancing tasks, but it represents another promising approach to the complex problem of dynamic stability control for biped robots.

Alongside the development of balance control techniques formulated and treated as constrained non-linear real-time optimization problems, differential dynamic program, deep neural networks and other methods, it has been researched whether the problem of whole-body balancing can be addressed solely with linear optimal control theory, mainly motivated by the conceptual simplicity and low computational requirements. The work presented in [15] showed promising results for the application of full dynamics LQR control to bipedal balancing under the influence of disturbances while standing and while tracking a squatting motion. The paper also highlights how the LQR-formalism automatically exploits the dynamic coupling of the different limbs and how this is reflected in the composition of the state-feedback gain matrix. One of the key findings was that the use of the linearized model yields a linear controller that is even valid when the robot state deviates significantly from the state of linearization. Balancing with the lower body of a humanoid was achieved with only one system linearization over the entire range of a squatting motion. Different control objectives were represented in the cost function: Joint position tracking ('LQR') and additionally tracking of the system's linear and angular momentum ('LQR-momentum'), where the momentum is expressed as linearly dependent on the system state. While it is reported that LQR-momentum was 'much easier to tune', the performance remains comparable to the generic whole-body LQR.

A follow-up study extended this work by adding the ability to switch contact situations (from double to single support and back), as is necessary for walking [16]. Only one system linearization and state feedback controller were necessary per contact situation during static walking motions, again suggesting that LQR control can achieve its stabilizing effect over a wide range within the state space. Gain matrices for the different contact situations reveal a coupling of the stance leg joints and largely independent position control of the swing leg. Experiments were conducted with the lower body of a torque-controlled robot and a 1DoF torso mock-up.

In both cases not much was reported on the tuning process of the LQR, where tuning means an optimal design choice of the weight matrices in the LQR cost function. For more complex control problems, bringing together LQR control and non-linear system dynamics requires systematic tuning of these weights.

A recent study on automatic LQR tuning for a robotic application has shown that tuning in the weight space of the LQR cost function can significantly improve controller performance for non-linear control systems [17]. The authors use weight matrices parameterized by a set of hyperparameters θ_n for the controller design ('design weights') and experimentally evaluate the actual cost using fixed weight matrices ('performance weights'). They use Entropy Search (ES) to efficiently search over the space of hyperparameters for a controller with minimal actual cost. The control problem they are looking at is essentially a planar inverted pendulum with a single input (linear acceleration of the base). While this is a compelling showcase for the feasibility of the method, it is a significantly simpler problem and the parameter space to optimize over is far smaller than in the setting of whole-body humanoid balancing that we are considering.

B. Contribution

In this work we seek to further leverage the capabilities of LQR whole-body control by finding optimal controllers in the space of entries of the weight matrices used to synthesize the controller (the design weight space). We explore new linearized models for the ground contact that improve LQR performance. We use a simplified four-link humanoid model, and a dynamic model of the ARMAR-4 humanoid robot [18] with actuated legs, arms and torso, considering the task of stabilizing the robot after a push along the sagittal plane. While the controller is synthesized based on the linearized robot and contact dynamics, the optimization objective is to find a controller that prevents the robot from falling under the largest possible push in a non-linear, accurate physics simulation.

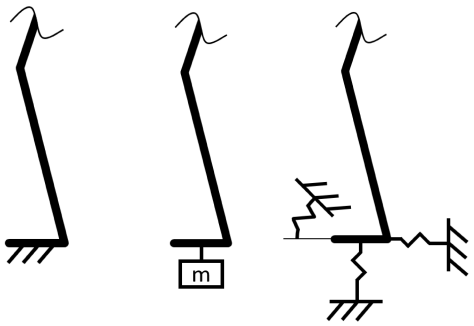
II. Approach

A. Linear ground contact modeling

Whereas the dynamic equations of motions formulated in the joint angles of the robot are relatively straight-forward to

linearize around a specific set-point (the *nominal posture*), the ground contact is more challenging to handle in linear LQR controller design. The real ground contact is unilateral, meaning that it cannot exert forces that pull the robot down. Horizontally, legged robots need to stay within the regime of static friction to prevent the feet from sliding. This requires the ground-contact force vector to remain within the friction cone, the opening angle of which is a variable of the foot and ground materials. Mathematically, this translates to a geometric inequality constraint, adding to the degree of contact non-linearity.

Since neither the unilaterality nor the friction cone constraint can be represented in a linear model required for LQR design, we need to find linear representations for these conditions. As modeling is the first part of the LQR design process, these consideration need to be made early on.



(a) Clamped (b) High Inertia (c) Springs

Fig. 2: Three different ground contact modeling approaches: (a) clamped to the ground, (b) high foot inertia and (c) springs.

We consider three different linear models for LQR design:

1) Clamped to the ground: The most straight-forward way of representing the ground contact is by having the feet clamped to the ground. This prevents any relative motion of the feet in the linear model but also allows the controller to apply arbitrarily high moments around the ankles without consequences. The so generated controllers therefore tend to rely on the 'ankle strategy', where the ankle joints are primarily used for balancing rather than developing coordinated whole-body motions that account for torque limits that, in reality, are imposed by the finite size of the feet.

2) High inertia feet: Another method that we are investigating here is to model the reaction forces between the ground and the robot at the foot contact as inertia forces acting on a very heavy foot. By increasing the weight of the foot, the inertia forces can become high enough to sufficiently prevent foot motion over one simulation time-step, even if the robot exchanges significant interaction forces and moments with the ground. To compensate gravity, we add appropriate constant vertical forces acting on each foot.

3) Springs: The third modeling method that we consider is to attach the robot's feet to the ground by means of springs along the three translational and around the three rotational axes. Those springs hold the feet in place and can naturally be modeled as linear components. Every translation or rotation of the foot results in proportional forces or moments.

The latter two methods have one major conceptual advantage over the clamped contact model in the context of modeling for LQR design: The LQR state vector in general does not include forces and torques, which means that the controller cannot take them into account during execution. By allowing the feet to actually move in the spring and inertia formulation, their displacement and the associated forces (from stiffness or inertia) are implicitly known to the controller and can be penalized. The high inertia model furthermore greatly simplifies the modeling for more than one ground contact (which simply requires assigning large weights and gravity-opposing forces to the end-effectors), which is a significant challenge in the case of the clamped model where the robot and the ground form a closed kinematic chain.

B. Design weight optimization

1) LQR design weights: An LQR full state feedback controller takes on the very elegant form of

$$u(t) = -K(x(t) - x_0) \tag{1}$$

where u is the computed input (the vector of joint torques in our case), K is the feedback gain matrix, x the system state (joint positions and velocities in our case) and x_0 the point of linearization. The matrix K is designed in such a way that both the input and the state space trajectory of the linear system minimize a cost J with

$$J = \int_0^\infty x^T(t)Qx(t) + u^T(t)Ru(t)dt \tag{2}$$

for the infinite horizon case, where Q and R are matrices containing the optimization weights used in the design process and will henceforth be called *design weights*. This underlying optimization can be solved by the solution of the (discrete) algebraic Riccati equation (DARE) and makes any so designed controller an optimal controller (for the linear system). For complex systems, the gain matrix K encodes the generally non-intuitive coupling terms between the multiple degrees of freedom, which makes it difficult to directly improve K. In contrast, the design weight matrices Q and R are much more approachable: It is common to set them up as diagonal matrices, so that every non-zero entry in Q directly corresponds to the penalization of a deviation from a specific desired state, and every entry in R penalizes the control effort of a specific input. Similar to [17] we therefore chose to optimize the entries of Q and R (design weight optimization) rather than the entries of K.

2) Domain-specific cost function: In a mechanical system with the state x containing joint positions and velocities, a controller as presented in Equation 1 designed with diagonal Q is an optimal state-feedback controller, but not necessarily a good balancing controller. We therefore need to adapt the state cost Q to give the controller a notion of stability.

We compose the state cost Q of three contributions: A diagonal matrix Q_A that motivates the controller to keep its initial pose (*posture cost*), a center of mass (CoM) matrix Q_B that motivates it to keep the CoM over its feet for static stability (*CoM cost*) and an angular momentum matrix Q_C, challenging the controller to minimize angular momentum for dynamic stability (*momentum cost*). Each of these matrices contains its own diagonal design weight matrices Q_1, Q_2 and Q_3 and together they form the overall state cost Q:

$$Q_A = Q_1 \tag{3}$$

$$Q_B = T_{CoM}^T Q_2 T_{CoM} \tag{4}$$

$$Q_C = T_H^T Q_3 T_H \tag{5}$$

$$Q = Q_A + Q_B + Q_C \tag{6}$$

T_{CoM} and T_H are the linearized mappings from the system state to the CoM position and the angular momentum around the CoM, respectively.

III. Experimental Setup

To enable the LQR-controller to regulate the highly non-linear system with the state cost computed from Equation 6 we optimize over a sub-space of the design weight space spanned by the entries of Q and R, or more precisely by the entries of Q_1, Q_2, Q_3 and R. The goal is to have the robot withstand the largest possible push.

A. Stochastic optimization

For each controller generated with the current design weights, this goal is evaluated and discretely quantified in a dynamics simulation that accurately reproduces the robot and contact non-linearities. The vertical contact force is produced by a virtual damped spring with a stiffness of $10^5 \frac{N}{m}$ and a damping coefficient of $10^4 \frac{Ns}{m}$ that is only active when the foot is in ground contact, and only exerts repulsive forces. Horizontal friction forces are generated by viscous damping of $1.5 \times 10^3 \frac{Ns}{m}$ when the foot is in ground contact.

The robot model initially stands in a statically stable pose and the torques computed by the LQR controller as well as gravity compensating torques are applied to its joints. It is then pushed from the front at hip height with a constant force for a fixed duration (0.5 s for the simplified model, 0.25 s for the ARMAR-4 simulation model). By varying this force we can vary the transmitted push impulse which we use as the defining parameter of the push. To evaluate the objective function of the maximization we perform successive simulations with iteratively increased push impulses (0.5 Ns for the simplified model, 5 Ns for the ARMAR-4 model) until the robot falls, taking the greatest impulse withstood by the robot as the current performance. While 0.5 Ns (or 5 Ns, respectively) is a comparatively coarse performance quantization, it allows us to run many evaluations to optimize over a wide range of the parameter space in a reasonable time span.

The optimization problem is particularly difficult for three reasons: (1) Neither the objective function (i. e. the mapping from design weights to controller performance) nor its gradient are known, but are known (from initial experiments) to have many local extrema. (2) The parameter space is so high-dimensional that any sampling in it (that can be accomplished in an acceptable time-frame) is inevitably sparse. (3) Sampling the objective function is costly since it requires running multiple experiments in the dynamic robot simulations.

These characteristics call for an efficient, sampling-based global optimization technique. In contrast to [17] where the cost function is explicitly modeled as Gaussian Process (GP) we chose a model-free optimization due to the high dimensionality of the parameter space. A stochastic global optimization technique that suits the problem at hand is *Simulated Annealing*. The optimization cycle is schematically depicted in Figure 3.

B. Simulated annealing

Simulated annealing is a physically inspired optimization meta-heuristic that stochastically samples the performance function over the parameter space, where the sampling is guided by a monotonously falling *temperature* parameter [19]. From an initial random location in the parameter space a new location in a certain distance is chosen and the performance function there is evaluated. Depending on whether the performance function in the new location is better or worse than in the current location, this new location is accepted as the new initial location. At the initial high temperature the algorithm is likely to jump to new locations even when they result in worse performance, to avoid getting stuck in local optima. As the temperature decreases with every cycle the algorithm only choses to stay in new locations that actually increase performance, eventually converging to a local optimum. To increase the algorithm's chances to find the global optimum, re-annealing, where the temperature is increased after the algorithm has consecutively chosen sample locations in close proximity to each other (i. e. has converged), can be applied.

1) Implementation Details: We start at an initial temperature $T_0 = 600$, leading to good initial coverage of the search space. The temperature T is lowered according to $T[t] = \alpha T[t-1]$ with $\alpha = 0.95$. At each temperature step, three optimization steps are performed. New design weights are sampled from a normal distribution around the current design weights. The admissible range for each weight parameter is from 1 to 10^7. Randomly chosen weight parameters outside this range are discarded, and the normal distribution is sampled again. If no improvement is made after 40 consecutive steps, re-annealing is applied.

C. Simplified 2D humanoid model

The simplified 2D model is similar to the human body in proportions and weight distribution, with a total mass of 7.8 kg. To enhance its balancing capability we derive three sets of linearized equations of motion (linear models) based on the three different contact models described in II-A and synthesize LQR controllers with the domain-specific cost

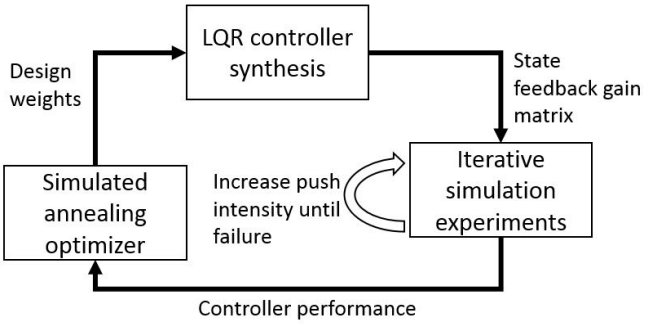


Fig. 3: Optimization in the LQR design process: The performance (maximum push impulse that can be recovered from) of the current controller is evaluated in a non-linear physics simulation. The performance is input to the optimizer that selects a new set of LQR design weights. A new controller is synthesized with those weights and the cycle repeats.

detailed in II-B.2. We chose an 18-dimensional subspace of the design weight space containing the six diagonal entries of the position feedback in Q, all six diagonal elements of R, the 2D position of the CoM, the 1D angular momentum and the three spring stiffnesses (in case of the spring ground contact model) as the search space for an optimal parameter set yielding the most robust controller.

D. Full 3D humanoid model

Considering all 63DoFs of the ARMAR-4 humanoid robot as well as the additional DoFs of the floating base is not feasible, as the search space of the optimization would become intractably large. We therefore limit our controller to the robot joints that presumably have the largest influence on the balancing performance, namely all leg joints, the two torso joints and two shoulder joints per arm. We further assign the same state cost coefficient to each of the three hip joints and take advantage of the robots symmetry, assigning the same coefficient to the left and right instances of the same joint. The full humanoid model has a total mass of 73.9 kg.

IV. Evaluation

The goal of the evaluation is twofold: First, we want to demonstrate that LQR controllers for whole-body balancing can be improved by optimizing in the design weight space using Simulated Annealing. Secondly, we want to show that the linearized representation of the ground contact plays a major role in the overall controller performance, which is reflected both in the quantitative and qualitative controller performance, i. e. the magnitude of pushes that can be withstood and the motion characteristics. The evaluation is based on a simplified 2D model, and the feasibility of the found approach for balancing the ARMAR-4 full body humanoid is demonstrated.

A. Simplified 2D humanoid model

From the initial position depicted in the first frame of Figure 6, the simplified model can be tipped over with a push of about 1 Ns applied from the front at the pelvis when all joints are locked in position. We consider this the baseline performance.

1) Contact model and optimization evaluation: We run 10,000 steps of Simulated Annealing for all three linear contact models, and evaluate the performance on the non-linear simulation model at every step. This optimization takes about two days for each model. Figure 7 shows the current best performance, measured by the impulse of the largest withstandable push, for the controllers based on the three different ground contact models. After 10,000 iterations the linear model based on the spring ground contact model yields the push-recovery LQR with the highest performance, enabling the model to cope with pushes of up to 5.5 Ns, a more than 500 % improvement in performance over the baseline. The controllers based on the other two contact models show similar but slightly worse performance.

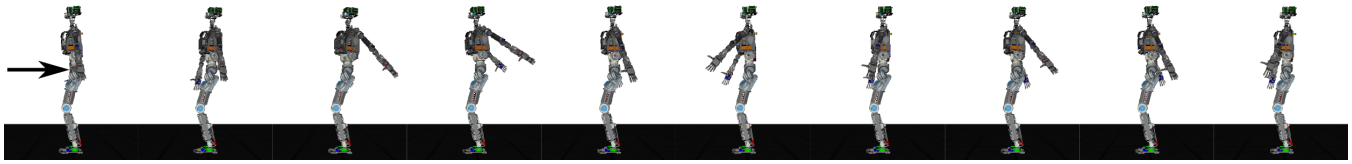
From these initial experiments we conclude that ground contact modeling with springs leads to the highest-performance controllers. We further conclude that Simulated Annealing is capable of finding good design weights for the LQR balance controller despite the difficulties described in III-A.

2) Independent contact model evaluation: To statistically evaluate the contact models independently of the optimization procedure we randomly pick 10,000 positions in the design weight space and synthesize and evaluate controllers based on all three contact models at these locations (essentially replacing the optimizer in Figure 3 by random selection). Figure 8 shows the accumulated results of these tests, where performances under 3.5 Ns are not included. Even under these purely stochastic conditions, controllers based on the spring contact model show better performance. The maximum performance level achieved with random parameters is 5 Ns, less than what could be achieved using Simulated Annealing.

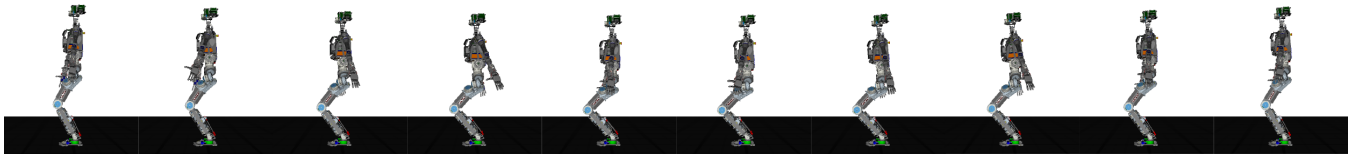
B. Full 3D humanoid model

Despite its lower performance as compared to the spring model (see Figure 7) we chose the high inertia ground contact model for the 3D evaluation. Consistently formulating the spring model for the case of the double contact is still part of our ongoing work. The high inertia model has the advantage of straight-forward extension to multi-contact situations.

1) Controller: Running 10,000 optimization steps of Simulated Annealing (requiring almost three weeks) leads to a capable LQR-controller that is able to stabilize the robot under the occurrence of frontal pushes of up to 29 Ns. A comparable experiment was presented in [15] where the "full humanoid model" withstood a push of 15 Ns, making much less use of the arms than our controller. The structure of the gain matrix K is visualized in Figure 9 and reveals the complex couplings between the different DoFs, represented by the non-zero off-diagonal elements. Capping the commanded joint torques at the robot's joint torque limits does neither influence the motion characteristics nor the controller performance noticeably.



(a) Simulation sequence of the 3D robot model reacting to a push from the front at hip height. It can be seen that the robot bends forward to shift its CoM and quickly swings its arms to compensate its angular momentum. The robot eventually comes back to its initial position.



(b) Simulation of the 3D robot moodel performing a squat under LQR balance control. No re-linearizations around any other than the initial state were performed along the trajectory.

Fig. 4: Simulation snapshots of ARMAR-4 balancing (a) under the influence of a push and (b) while performing a squat. The same linear whole-body controller was used in both experiments. Images from left to right with 0.3 s temporal spacing.

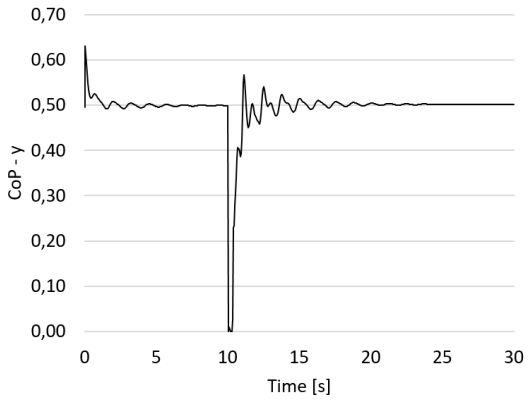


Fig. 5: Center of Pressure during a 30 s balancing experiment in relative foot coordinates (heel≙0, toes≙1). A push is applied at $t = 10$ s. The corresponding robot motion after the push is shown in Figure 4 (a).

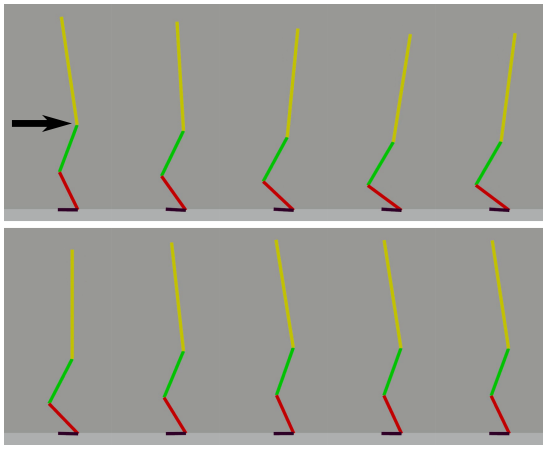


Fig. 6: The simplified four-link 2D humanoid model performing whole-body balancing after being pushed from the front. Consecutive frames are spaced 0.2 s apart.

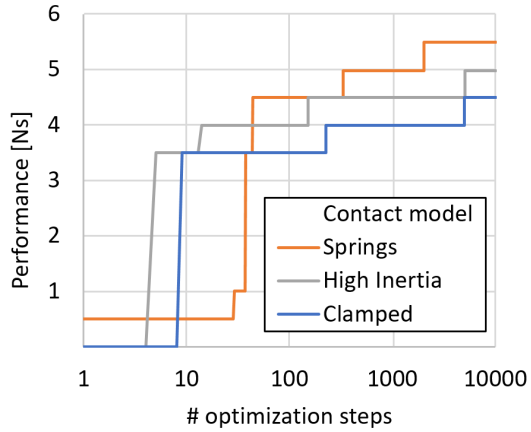


Fig. 7: Performance evolution over the course of 10.000 optimization steps with Simulated Annealing for push-recovery LQRs based on the three different ground contact models.

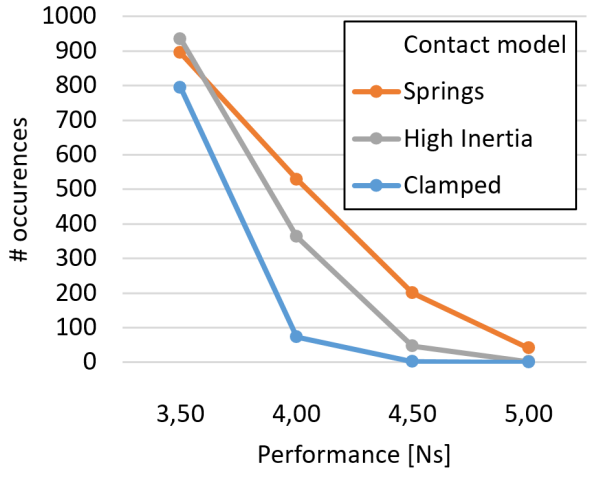


Fig. 8: Controller performance based on the three different contact models evaluated over 10.000 random positions in the parameter space. Depicted are the accumulated performance evaluations for each contact model, excluding performances under 3.5 Ns.

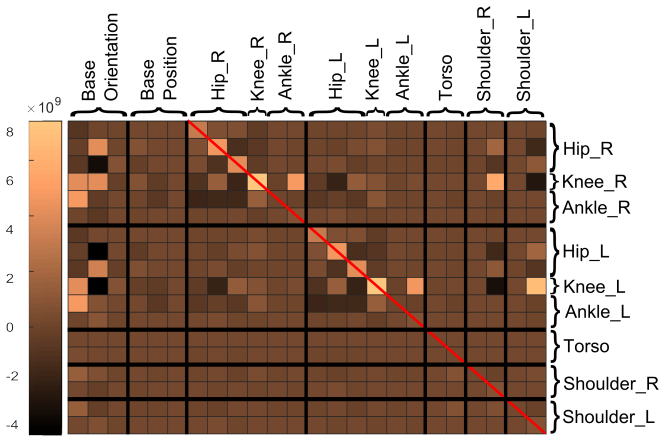


Fig. 9: LQR position feedback matrix for the 3D whole-body balance controller, showing how the torques for the 18 actuated joints (right) are computed from the position errors of the 24 considered DoFs (top), including the unactuated floating base. Any non-zero entries off the diagonal (red) represent the interconnections between the joints, such as the clearly visible coupling between the shoulder and the hip joints. The velocity feedback matrix has a very similar structure.

2) Motion characteristics: Figure 4(a) shows a simulation experiment in which the humanoid is pushed from the front and reacts by bending its upper body to change its CoM position as well as throwing its arms backward to compensate its angular momentum. We argue that this holistic motion is a benefit of the way we model the ground contact, as the controller is given a motivation to apply less torque around the ankle joints and do more with the upper body. The position of the center of pressure (CoP) within the feet is depicted Figure 5: It travels all the way to the heel directly after the push but is quickly brought back to the center of the foot by the compensatory motion of the robot.

3) Trajectory tracking: The same controller can be used to track a squatting motion by commanding a vertical sinusoidal pelvis trajectory with an amplitude of 20 cm (see Figure 4(b)). The only difference to the standing case controller from Equation 1 is that the control error is computed with respect to the desired joint trajectories $x_d(t)$ and not to the initial, time-invariant nominal posture, yielding

$$u(t) = -K(x(t) - x_d(t)) \tag{7}$$

The controller can not only track the trajectory but also compensate significant pushes that are applied during squatting by bending its torso and swinging its arms. No re-linearization along the squatting trajectories was used.

4) Robustness against model uncertainties: We tested the ARMAR-4 controller with a simulation model in which certain links in the legs and arms had increased mass parameters of up to 10%, that were not accounted for in the controller optimization. Balance control performance thereby remained on a comparable level as without alterations, indicating that weight-optimized LQR controllers have a certain robustness against parameter uncertainties and are suitable to be transferred from simulation environments (where they can be optimized without putting the robot at risk) to real hardware, where the exact model parameters might not be known.

V. Conclusion

We present our work on applying the LQR control method to the problem of torque-based whole-body balancing for humanoid robots. We focus on handling the non-linearities of the robot by developing new linear ground contact models and by systematically optimizing the LQR design weights. The two main novelties of this work are (1) the proposition and evaluation of two new linear ground contact models (high inertia, springs) and (2) the application of Simulated Annealing for optimizing the LQR design weights with respect to the overall controller performance (i. e. resilience against pushes). We explicitly take into account the arms and include the CoM position in the control objective, leading to a balance controller that makes intuitively plausible use of the robot's arms to keep its balance (see Figure 4).

We show that the ground contact model used in the controller synthesis has a significant influence on the controller performance and on the quality of the compensatory motion in the physically consistent robot simulation, and that our newly proposed models lead to improved use of the arms for balancing as compared to comparable methods found in the literature. We further showed that the application of Simulated Annealing for tuning the balance LQR in the design weight space consistently improves performance, for the two different robot models as well as for the different ground contact models (see Figure 7 and Figure 8), improving the whole-body balance performance of the simplified test model by over 500%.

Our results suggest that LQR control can be an effective control method for whole-body balancing, and that it can be improved by better contact models and systematic tuning with stochastic optimization techniques. Our experiments on controlling a squatting motion with a controller based on a single linearization substantiate the suggestion made in [15], [16] that the region of applicability of a single LQR balancing controller extends far from the initial point of linearization, and our experiments with an altered robot model suggest a certain robustness of the controller against model uncertainties.

A. Limitations

Owing to its computational simplicity that makes full-dynamics LQR state-feedback control so attractive, there are a few relevant limitations in its capabilities in the context of whole-body balancing. Most notably, joint position limits, joint torque limits and contact friction constraints cannot be considered explicitly (other than by limiting the computed output). They can only be implicitly encoded in the state and actuation cost. Our proposed ground contact models allow for a more effective consideration of the friction limits as they make the contact forces implicitly known to the controller, and thereby penalizeable in the state cost matrix.

B. Future Work

Future work will aim at quantifying the volume of the state space in which one linearization and one optimized controller is applicable. The goal is to find a minimal set of LQR controllers that can stabilize a robot over its entire range of (dynamic) motions.

References

[1] S. Kudoh, T. Komura, and K. Ikeuchi, "The dynamic postural adjustment with the quadratic programming method," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3. IEEE, 2002, pp. 2563–2568.

[2] B. J. Stephens and C. G. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1248–1255.

[3] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[4] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Optimization based controller design and implementation for the atlas robot in the darpa robotics challenge finals," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 1028–1035.

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

[6] M. Posa, S. Kuindersma, and R. Tedrake, "Optimization and stabilization of trajectories for constrained dynamical systems," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1366–1373.

[7] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[8] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.

[9] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*. IEEE, 2006, pp. 200–207.

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

[11] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *ICRA*, vol. 3, 2003, pp. 1620–1626.

[12] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS international conference on humanoid robots*, 2006.

[13] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4906–4913.

[14] Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. van de Panne, "Feedback control for cassie with deep reinforcement learning," *arXiv preprint arXiv:1803.05580*, 2018.

[15] S. Mason, L. Righetti, and S. Schaal, "Full dynamics lqr control of a humanoid robot: An experimental study on balancing and squatting," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 374–379.

[16] S. Mason, N. Rotella, S. Schaal, and L. Righetti, "Balancing and walking using full dynamics lqr control with contact constraints," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 63–68.

[17] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe, "Automatic LQR tuning based on Gaussian process global optimization," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 270–277.

[18] T. Asfour, J. Schill, H. Peters, C. Klas, J. Bucker, C. Sander, S. Schulz, A. Kargov, T. Werner, and V. Bartenbach, "ARMAR-4: A 63 DOF torque controlled humanoid robot," in *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2013, pp. 390–396.

[19] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.