Increased Robustness of Humanoid Standing Balance in the Sagittal Plane through Adaptive Joint Torque Reduction

Markus Giftthaler[1] and Katie Byl[2]

***Abstract*— This paper introduces and compares two control approaches that increase the robustness of humanoid standing balance in the sagittal plane with respect to impulsive perturbations by adapting joint torques. To address the question of how the range of admissible perturbations for an n-link inverted pendulum model can be enlarged, we propose two different strategies: adapting the ankle torque only and adapting all joint torques uniformly. For each, explicit-form solutions exist for nonlinear models with an arbitrary number of links. A Center of Pressure-based criterion for switching between the default feedback controller and the torque reduction strategies is introduced. In a three-link model case study, a wide range of robot poses, which are optimized either for steady state effort minimization or robustness, are considered. Simulation results show that our models are robust to impulse perturbations of between 10% and 149% greater magnitude than for an LQR default control law. However, there is a trade-off between robustness gains and steady-state balancing effort. In a second example, a joint-locked model, which uses an adaptive joint torque reduction strategy, outperforms a joint-unlocked model that only uses the default controller both in terms of robustness and control effort.**

I. INTRODUCTION

The technological advances in humanoid robotics in recent years have driven our understanding of bipedal robots towards autonomously operating machines with a broad spectrum of possible applications. Considering such a robot which is standing in place, it is desirable that the system is able to perform tasks like object manipulations without destabilizing balance, i.e., without forcing the system to readjust its footholds. Therefore, uncertainty and robustness are important issues in balance research. Because they facilitate both modeling and physical understanding, one- and multi-link inverted pendulum models have proven useful for balance studies and walking control for both humans and humanoid robots [1] [2]. Well-known examples for balancing strategies are the ankle and hip strategies [3]. Investigations about their robustness properties exist, e.g. in [4].

In this paper, we are considering the control of a low-dimensional humanoid robot model in the sagittal plane. Common approaches are Parametric State Feedback controllers, Feedback-Linearization techniques or Linear Quadratic Regulators (LQR) [5]. Optimization based methods using Dynamic Programming and look-up tables have been presented in [6]. An approach using Model Predictive Control is available in [7]. While optimization related methods usually exhibit good robustness properties, they require a discretization of the state space, high computational efforts, and/or the generation of large look-up tables. Recently, techniques that modify controller feedback gains depending on the disturbance have come to the interest of the research community. Xing, Atkeson and others have presented results showing that postural feedback gains should change with perturbation size [8]. In [9], it is shown that the range of admissible perturbations can be enlarged by adapting the knee joint torque to the ankle torque, based on Reaction-Null-Space methods. However, such control policies cannot be generalized for an arbitrary number of links easily, nor are they available in explicit form.

The rest of this paper is organized as follows. We present the model of our robot in Section II. In Section III, we introduce a robustness improving control approach for upright balancing: it is based on systematic adaption of the robot joint torques to the magnitude of an impulsive perturbation. In Section IV, we present simulation results that show robustness gains for different robot poses and also a reduction of control effort. Conclusions are drawn in Section V. A discussion about the presented material and future work is given in Section VI.

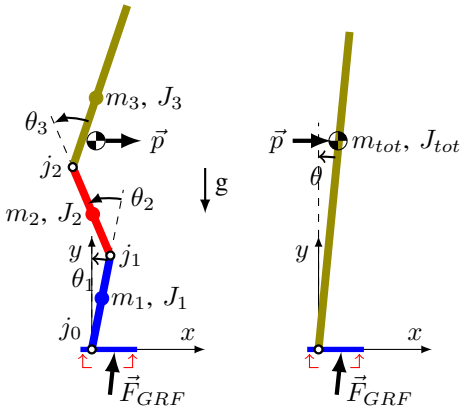


Fig. 1: Three-link and one-link inverted pendulum models in the sagittal plane. All angles are zero when the robot stands fully upright.

II. MODEL

Fig.1 shows a three-link and a one-link inverted pendulum robot model in the sagittal plane, both facing to the right. Each robot is assumed to be standing on a flat, massless, asymmetric foot. As for the actuators, we consider torque

[1]M. Giftthaler is with the Department of Mechanical and Process Engineering, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland. gimarkus at ethz.ch

[2]K. Byl is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106, USA. katiebyl at ece.ucsb.edu

TABLE I: Three link robot model parameters

No.	Link	Mass m_i [kg]	Inertia J_i [kg·m^2]	l_i[m]	$l_{c,i}$[m]
1	Shanks	8.934	0.163	0.422	0.230
2	Thighs	17.093	0.311	0.424	0.199
3	Torso	61.990	3.420	0.752	0.322

inputs at the joints. Table I shows the parameters of the three-link model, which are extracted from the DARPA Robotics Challenge Atlas humanoid simulation [10]. Note that l_i denotes the length of link i and $l_{c,i}$ denotes the distance from joint j_{i-1} to the location of the center of mass m_i. J_i is the moment of inertia. The equations of motion for an n-link inverted pendulum can be derived using Lagrange's formalism, choosing the generalized coordinates $q = [\theta_1\, \theta_2 \dots \theta_n]^\top$ and can be written as

$$M(q)\ddot{q} + H(q, \dot{q}) + \Phi(q) = \vec{\tau} \tag{1}$$

with inertia matrix $M(q) \in \mathbb{R}^{n\times n}$, $H(\dot{q}, q) \in \mathbb{R}^{n\times 1}$ capturing the Coriolis and centrifugal forces, $\Phi(q) \in \mathbb{R}^{n\times 1}$ describing gravity-induced terms and $\vec{\tau} = [\tau_1\ \tau_2 \dots \tau_n]^\top \in \mathbb{R}^{n\times 1}$ being the vector of input torques.

For the purpose of this paper, we restrict the notion of robustness to the ability to recover from impulsive perturbations. It is assumed that an impulse $\vec{p}$ is imparted at the overall Center of Mass (CoM) while the robot is in a steady state. It occurs instantaneously, in the positive (or negative) x-direction only, and is entirely absorbed by the robot. This leads to a new distribution of angular velocities depending on the initial state and the impulse, if the robot joints are unlocked.

In case of the upper $n-1$ joints being locked, the robot model can be lumped into a corresponding one-link inverted pendulum with angle θ and identical coordinates of the CoM, which is also shown in Fig. 1. In that case, an imparted impulse leads to a single angular velocity $\dot{\theta}$.

The Ground Reaction Force (GRF) is a function of the current state and the control input and its location on the ground is called the Center of Pressure (CoP). We define an arbitrary Region of Trust (RoT) on the x-axis as the interval $[-0.04\ 0.18]$ (meters), in which the CoP shall be moved in order to recover from perturbations. The RoT is indicated with ↾ ↿ in Fig. 1. Note that the RoT is a subarea of the actual support polygon, which is defined to range from -0.05 m to 0.2 m. Thus, we want to prevent the CoP from coming close to the edge of the foot and provide a safety margin.

For the following investigations, we choose the underlying main control laws to be full-state feedback LQR controllers that are designed about certain reference poses, which we will refer to as our default control law. For the feedback-gain matrix computation, we choose all weighting matrices to be diagonal, penalizing deviations from the reference pose with factors 1 and control inputs with factors 0.1. There are no penalties for angular velocities. Note that in this setup, the choice of the weighting factors is not very influential because the feedback gains converge to the same matrix for a broad spectrum of different weighing factors.

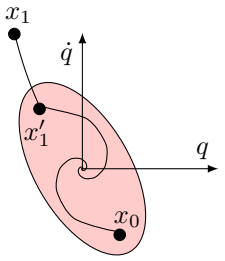


Fig. 2: A system starting at x_0 in the default controller's Basin of Attraction (BoA) converges to the setpoint. An Adaptive Joint Torque Reduction strategy can be used to drive a system starting with initial state x_1 into the BoA. The default controller takes over at x_1'.

III. THE ADAPTIVE JOINT TORQUE REDUCTION STRATEGY

A. Approach

Humans typically use a strategy of moving the CoP to recover from perturbations of small magnitude. The size of the support polygon imposes limits for admissible perturbations for this strategy. If required, different control strategies, for example taking a step or applying an external force, e.g. by pushing on a wall, have to be used [11]. In this paper, we assume the foot does not slide or lift off. We define it as a failure if the required CoP for a given control strategy would exit the RoT. Hence, our goal is to modify the balancing control strategy such that the maximum admissible perturbation is enlarged while the CoP stays in the RoT. This increases the bound for impulsive disturbances the robot can take without switching to higher-level control strategies.

Consider a Basin of Attraction (BoA) of a closed-loop inverted pendulum with a default controller and a 'Moving the CoP' strategy, as schematically shown in Fig. 2. Consider a perturbation which exceeds the described limit and thus leads to a state x_1 which lies outsides the BoA. We propose to drive the system's state back into the BoA by adapting (reducing) the joint torques calculated by the default controller such that the CoP stays at the edge of the RoT. Since the BoA's precise boundaries for a nonlinear system will in general not be computable with low effort, we suggest the controller switching strategy shown in Fig. 3. We estimate the required location of the CoP for the default-controller case. If it lies outside the RoT, the joint torques are adapted such that the CoP is placed at the edge of the RoT. We later switch back to the default controller if its estimated CoP location is back within the RoT.

In this paper, we propose two different ways of adapting the joint torques in order to place the CoP at a desired location:

I Reduce the ankle torque τ_1 as a function of the state and the default controls which remain unchanged for the upper $n-1$ joints. There exists a closed form solution; its derivation is outlined in Section III-B. In the following, we refer to this strategy as Ankle Torque Reduction (ATR).

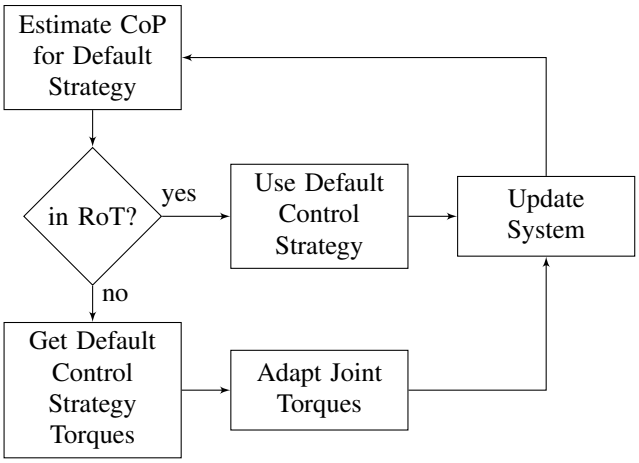


Fig. 3: Flow chart describing the switching strategy between using the default controller and using adapted joint torques.

II Reduce all n torques by a factor α, i.e., $\vec{\tau}_{\text{red}} = \alpha \cdot \vec{\tau}$. A similar derivation of a closed form solution exists but it is not included here, due to space limitations. In the following, we refer to this strategy as Uniform Joint Torque Reduction (UTR).

B. Derivation of the adapted ankle torque τ_1

The system state $\vec{q}$, the desired location of the CoP ξ_x and the default control inputs $\vec{\tau}$ are given. While the CoP is a function of τ_1 and the y-component of the GRF, $\vec{F}_{GRF}$ is itself a function of the state and the input torque vector. The following section sketches the derivation of the adapted ankle torque $\tau_1 = f(\vec{\tau}, \vec{q}, \xi_x)$ for the ATR strategy.

1) Calculation of the GRF and the CoP: Consider a general n-link system and let $\vec{p}_t$ be the total impulse of the system. Let $\vec{r}_a^{\,b}$ denote a vector from point a to b. Let the absolute angular velocity of link i be ω_i with corresponding angular velocity vector $\vec{\omega}_i = [0\ 0\ \omega_i]^\top$. The GRF $\vec{F}_{GRF} = [f_x\ f_y\ f_z]^\top$ can be computed from the conservation of impulse $\frac{d\vec{p}_t}{dt} = \sum_{i=1}^{n} m_i \vec{a}_{m_i} = \vec{F}_{GRF} - \vec{F}_g$, where $\vec{F}_g = [0 \quad g\sum_{i=1}^{n} m_i \quad 0]^\top$ represents the body weight and $\vec{a}_{m_i}$ denotes the acceleration of mass m_i. Hence the GRF can be written in matrix-vector form as

$$\vec{F}_{GRF} = \underbrace{[\vec{a}_{m_1}\ \vec{a}_{m_2} \dots \vec{a}_{m_n}]}_{=:A \in \mathbb{R}^{3\times n}} \underbrace{\begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}}_{=:\vec{m} \in \mathbb{R}^{n\times 1}} + \vec{F}_g \tag{2}$$

which simplifies the following derivation. Considering the equilibrium of torques about the ankle joint, the x-coordinate of the CoP ξ_x results as a function of the ankle torque τ_1 and the vertical component of the GRF f_y:

$$\xi_x = \frac{\tau_1}{f_y} \tag{3}$$

2) Define matrices Ω and $\dot{\Omega}$: For the n-link case, the derivative $\dot{q}$ of the generalized coordinates is mapped to the absolute angular velocities by a regular matrix $\mathcal{L} \in \mathbb{R}^{n\times n}$ such that $[\omega_1 \dots \omega_n]^\top = \mathcal{L}\dot{q}$ and $[\dot{\omega}_1 \dots \dot{\omega}_n]^\top = \mathcal{L}\ddot{q}$. Consider a matrix $\Omega = [\vec{\omega}_1 \dots \vec{\omega}_n] \in \mathbb{R}^{3\times n}$ and its derivative w.r.t. time $\dot{\Omega}$. Using the equation of motion (1) for the sagittal plane, we can rewrite $\dot{\Omega}$ as

$$\dot{\Omega} = \begin{bmatrix} \vec{0}^\top \\ \vec{0}^\top \\ -(\mathcal{L}M^{-1}(H+\Phi))^\top \end{bmatrix} + \begin{bmatrix} \vec{0}^\top \\ \vec{0}^\top \\ (\mathcal{L}M^{-1}\vec{\tau})^\top \end{bmatrix} \tag{4}$$

as a first step to isolate the torque vector $\vec{\tau}$.

3) Isolating the ankle torque τ_1: Rewriting the matrix $\mathcal{L}M^{-1} =: [\vec{\eta}_1\ \vec{\eta}_2 \dots \vec{\eta}_n]$, $\vec{\eta}_i \in \mathbb{R}^{n\times 1}$ allows us to isolate the ankle torque from the second summand of Equation (4)

$$\mathcal{L}M^{-1}\vec{\tau} = \tau_1\vec{\eta}_1 + \mathcal{L}M^{-1} \cdot \underbrace{[0\ \tau_2 \dots \tau_n]^\top}_{=:\vec{\tau}^{*\top}} \tag{5}$$

which leads to

$$\dot{\Omega} = \underbrace{\begin{bmatrix} \vec{0}^\top \\ \vec{0}^\top \\ (\mathcal{L}M^{-1}(\vec{\tau}^* - H - \Phi))^\top \end{bmatrix}}_{=:\dot{\Omega}_A} + \tau_1 \underbrace{\begin{bmatrix} \vec{0}^\top \\ \vec{0}^\top \\ \vec{\eta}_1^\top \end{bmatrix}}_{=:\dot{\Omega}_B} \tag{6}$$

where the first summand is just a function of the state and the input torques $\tau_2 \dots \tau_n$ which are set by the default control law. The second summand now contains the unknown variable τ_1 as simple, scalar factor.

4) Calculation of the acceleration of every m_i: The acceleration of an arbitrary point p_i on link i is

$$\vec{a}_{p_i} = \vec{a}_{j_{(i-1)}} + \dot{\vec{\omega}}_i \times \vec{r}_{j_{i-1}}^{\,p_i} + \vec{\omega}_i \times (\vec{\omega}_i \times \vec{r}_{j_{i-1}}^{\,p_i}) \tag{7}$$

with j_{i-1} denoting the preceding joint. The cross products can be expressed as matrix-vector products, similar to $\vec{\omega}_i \times \vec{r} = \hat{\vec{\omega}}_i \cdot \vec{r}$ with $\hat{\vec{\omega}}_i \in \mathbb{R}^{3\times 3}$ being a skew-symmetric matrix. This leads to the equivalent equation

$$\vec{a}_{p_i} = \vec{a}_{j_{(i-1)}} + \hat{\dot{\vec{\omega}}}_i \cdot \vec{r}_{j_{i-1}}^{\,p_i} + \hat{\vec{\omega}}_i^2 \cdot \vec{r}_{j_{i-1}}^{\,p_i}\ , \tag{8}$$

and eliminates the cross-product notation. Evaluating Equation (8) for a mass m_i by subsequent substitution delivers

$$\begin{aligned} \vec{a}_{m_i} = &\sum_{k=1}^{i-1} \hat{\dot{\vec{\omega}}}_k \cdot \vec{r}_{j_{k-1}}^{\,j_k} + \hat{\dot{\vec{\omega}}}_i \cdot \vec{r}_{j_{i-1}}^{\,m_i} \\ &+ \underbrace{\sum_{k=1}^{i-1} \hat{\vec{\omega}}_k^2 \cdot \vec{r}_{j_{k-1}}^{\,j_k} + \hat{\vec{\omega}}_i^2 \cdot \vec{r}_{j_{i-1}}^{\,m_i}}_{=:\vec{e}_i} \end{aligned} \tag{9}$$

with $\vec{e}_i$ being a function of the current state only. Aiming at a matrix-vector representation for the acceleration matrix A, we now define a new matrix R to be

$$R = \begin{bmatrix} \vec{r}_0^{\,m_1} & \vec{r}_0^{\,j_1} & \vec{r}_0^{\,j_1} & \dots & \vec{r}_0^{\,j_1} \\ 0 & \vec{r}_{j_1}^{\,m_2} & \vec{r}_{j_1}^{\,j_2} & \dots & \vec{r}_{j_1}^{\,j_2} \\ 0 & 0 & \vec{r}_{j_2}^{\,m_3} & \dots & \vec{r}_{j_2}^{\,j_3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \vec{r}_{j_{n-1}}^{\,m_n} \end{bmatrix} \in \mathbb{R}^{3n\times n} \tag{10}$$

This allows us to write the acceleration matrix A as

$$A = [\hat{\dot{\vec{\omega}}}_1 \; \hat{\dot{\vec{\omega}}}_2 \ldots \hat{\dot{\vec{\omega}}}_n] \cdot R + \underbrace{[\vec{e}_1 \; \vec{e}_2 \ldots \vec{e}_n]}_{=:E \in \mathbb{R}^{3\times n}} \tag{11}$$

by exploiting the linear structure of Equation (9).

5) Equating the results: The advantage of the arrangement in Equation (11) is that the angular acceleration vectors which correspond to the skew-symmetric matrices $\hat{\dot{\vec{\omega}}}_1 \ldots \hat{\dot{\vec{\omega}}}_n$ can be identified as the column vectors of $\dot{\Omega}$ from Equation (6). For both $\dot{\Omega}_A$ and $\dot{\Omega}_B$ we replace each column vector by its corresponding skew-symmetric matrix and obtain $\dot{\Omega}_A \to \hat{\dot{\Omega}}_A \in \mathbb{R}^{3\times 3n}$ and $\dot{\Omega}_B \to \hat{\dot{\Omega}}_B \in \mathbb{R}^{3\times 3n}$. Substituting them into Equation (11) results in $A = \hat{\dot{\Omega}}_A R + \tau_1 \hat{\dot{\Omega}}_B R + E$ and substituting this in Equation (2) finally delivers

$$\vec{F}_{GRF} = \tau_1 \hat{\dot{\Omega}}_B R\vec{m} + (\hat{\dot{\Omega}}_A R + E)\vec{m} + \vec{F}_g \tag{12}$$

6) Solving for the adapted ankle torque: For a robot model with an ideally flat foot, only the y-component of the GRF and therefore the second row of Equation (12) is relevant for an equlibrium of torques about the ankle. It is useful to split some matrices along their rows:

$$\hat{\dot{\Omega}}_A = \begin{bmatrix} \vec{\varphi}_1^\top \\ \vec{\varphi}_2^\top \\ \vec{\varphi}_3^\top \end{bmatrix}, \; \hat{\dot{\Omega}}_B = \begin{bmatrix} \vec{n}_1^\top \\ \vec{n}_2^\top \\ \vec{n}_3^\top \end{bmatrix}, \; E = \begin{bmatrix} \vec{\epsilon}_1^\top \\ \vec{\epsilon}_2^\top \\ \vec{\epsilon}_3^\top \end{bmatrix} \tag{13}$$

The second row of Equation (12) can now be solved for the ankle torque τ_1 as a function of the default control inputs for the remaining $n-1$ joints and the desired location of the CoP ξ_x by using Equation (3). This leads to the closed-form expression

$$\tau_1 = \frac{(\vec{\varphi}_2^\top R + \vec{\epsilon}_2^\top)\vec{m} + g\sum_{i=1}^{n} m_i}{\xi_x^{-1} - \vec{n}_2^\top R\vec{m}} \,. \tag{14}$$

Equation (14) can easily be evaluated numerically using matrix-vector operations.

IV. SIMULATION RESULTS

A. Choosing steady state robot poses based on different optimization criteria

First, consider the three-link case. A humanoid robot can take a large variety of different poses for balancing. Reflecting the requirement of being able to balance in both straight upright configurations as well as crouched poses, we chose to parameterize different sets of poses by the height of the Center of Mass $y_{\text{CoM}} \in [0.56 \; 0.96]$ (meters). Furthermore, x_{CoM} had to be within the RoT in steady state. Considering the space of kinematically feasible pairs of $(x_{\text{CoM}}, y_{\text{CoM}})$, we compared robustness towards impulsive perturbations depending on a given combination of

- y_{CoM}, the height of the CoM.
- the control strategy (LQR, ATR, UTR).
- poses being chosen according to steady-state balancing effort minimization, or LQR, ATR, or UTR robustness optimization.

For optimizing the pose for minimal steady state effort, Equation (1) delivers $\vec{\tau}_{ss} = \Phi$. We chose the sum of squared input torques as optimization criterion. This can be interpreted as a measure for power: joint torques are approximately directly proportional to the motor currents and the power is directly proportional to the sum of squared motor currents. We formulate the following problem for a desired $y_{\text{CoM}}^{\text{des}}$:

$$\begin{aligned} \min_q (\Phi^\top \Phi) \quad \text{s.t.} \quad & y_{\text{CoM}}(q) = y_{\text{CoM}}^{\text{des}} \\ & 0 \le \theta_1 \le \pi/2 \\ & 0 \le \theta_j \le \pi, \quad j = 2,3 \\ & 0 \le \theta_1 - \theta_2 + \theta_3 \end{aligned} \tag{15}$$

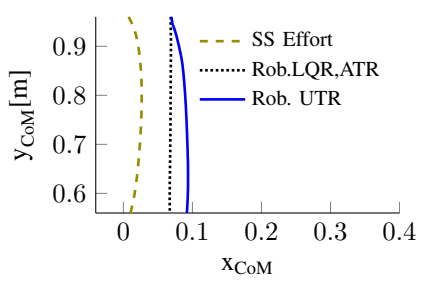
with the inequality constraints reflecting expected joint constraints for balancing humanoids. In a typical scenario for humanoid robot operation, one may be interested in choosing the robot pose for maximal robustness towards unexpected perturbations from both the front and the rear, for a given control strategy. Let p^+ (p^-) be the largest positive (negative) impulse a system can take in a particular pose without failing. Furthermore, consider the cost function $\mathcal{J} = (p^+ + p^-)^2$. If the bounds for negative and positive impulses are equal in magnitude, $\mathcal{J}$ results in zero. Otherwise it increases directly proportional to the square of the impulses' difference in magnitude. Therefore, it penalizes asymmetric perturbation bounds, or in other words, a lack of robustness in one direction compared to the other. The optimal poses for LQR, ATR, and UTR robustness w.r.t. the cost function $\mathcal{J}$ then arise from similar optimization problems as in Equation (15). These nonlinear optimization problems were solved using the Global Search Algorithm from the MATLAB Global Optimization Toolbox [12]. The LQR controller design was part of the optimization and solved at each iteration.

Different CoM locations result from the different optimization criteria as shown in Fig. 4a: while the steady state effort minimizing CoM location is relatively close over the ankle joint in x, the robustness optimizing solutions are closer over the middle of the asymmetric foot. Note that for both using an LQR controller only and using the ATR strategy, the same CoM locations result within the resolution of our numerical simulations. If we optimize the pose for the UTR strategy and robustness, the CoM location starts at the same point as for LQR/ATR at the upper end of the range of y_{CoM}, then it diverges slightly towards higher x_{CoM} values. Fig. 4b shows a series of stick figures which illustrate the different optimal steady-state poses for different heights of the CoM and different control strategies.

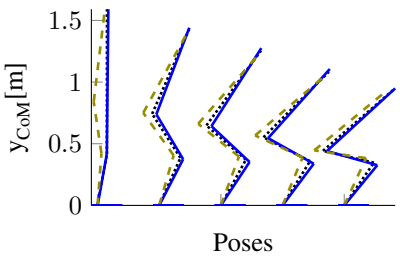
B. Numerical simulations show an increase of robustness using ATR and UTR strategies

Fig. 5 shows an example for ATR and UTR strategy simulation results for a system where the pose was chosen for steady-state effort minimization. The disturbance was chosen such that the state is driven out of the default controller's Basin of Attraction, consequently the LQR controller would fail. Using the joint torque reduction strategies, the CoP is controlled at exactly the edge of the RoT until the system reenters the BoA of the default controller.

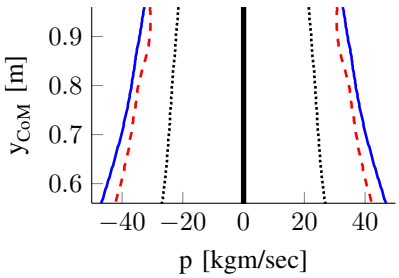
Fig. 4d - 4f show positive and negative perturbation bounds



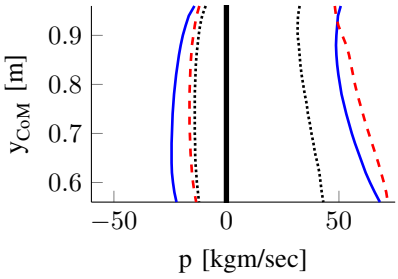
(a) CoM locations for the different pose optimization criteria steady-state effort minimization and robustness opt. for LQR/ATR and UTR.



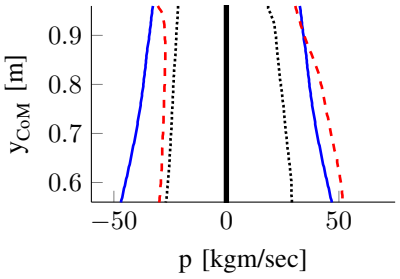
(b) A series of poses corresponding to the CoM trajectories in (a). Legend as in (a).



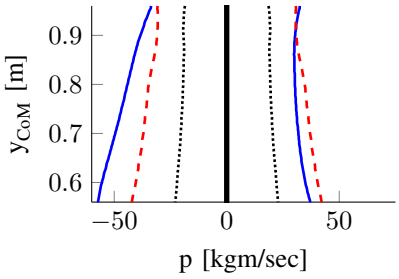
(c) Perturbation Bounds for LQR, ATR, and UTR about the pose which is optimized for robustness for each particular strategy.



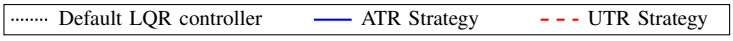
(d) Perturbation bounds for LQR, ATR, and UTR strategies: poses chosen for steady state effort minimization.



(e) Perturbation bounds for LQR, ATR, and UTR strategies: poses chosen for robustness optimization of LQR/ATR strategies.



(f) Perturbation bounds for LQR, ATR, and UTR strategies: poses chosen for robustness optimization of UTR strategy.



(g) legend for subfigures (c)-(f)

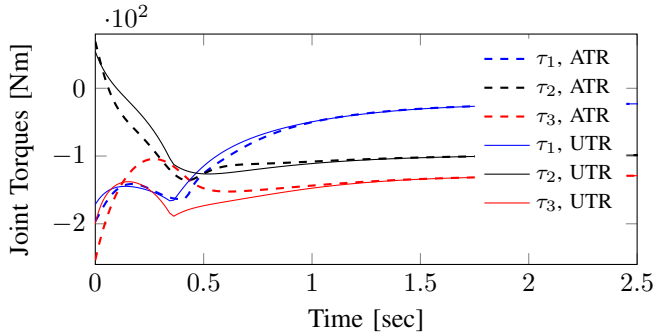
Fig. 4: Simulation results for different pose optimization criteria and control strategies.

for three-link models with different setpoint poses chosen according to the optimization criteria from Section IV-A and the control strategies LQR, ATR, and UTR. Both the ATR and UTR strategy always outperform LQR for all poses, for both negative and positive perturbations. Fig. 4d shows the perturbation bounds if the pose is chosen for steady state effort minimization. Note that the slight bend of the optimal CoM locations is reflected by perturbation bounds which are bent the other way round. The relative robustness gains lie between 10% in comparison to LQR in the worst case and 77% more robustness in the best case. If we choose the pose to be optimized for ATR/LQR robustness, the bounds for the admissible impulses on the CoM result as in Fig. 4e. Again, the smallest gain is 10%, the largest gain is 77%. The largest overall increase in robustness can be found for the case where the pose is optimized for robustness with the UTR strategy, see Fig. 4f, and is 149%.

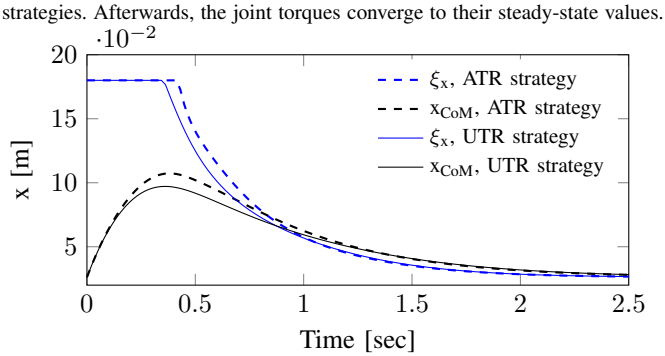
To summarize, in all three scenarios, the robustness of the default LQR controller is inferior to both presented Joint Torque Reduction strategies. However, at the same time, ATR is more robust w.r.t. negative disturbances than UTR. On the other hand, UTR is more robust towards positive disturbances than ATR, except for a small interval of y_{CoM} being between approximately 0.96 and 0.90.

Fig. 4c shows a plot which compares the perturbation bounds of LQR, ATR, and UTR, using the optimal x_{CoM} locations particular to each strategy and y_{CoM} height. Note that the curves are symmetric, due to the choice of the robustness cost function $\mathcal{J}$ in Section IV-A. In summary, UTR achieves between 39% and 57% more robustness compared to the default LQR controller, ATR gains between 51% and 74%. That means, if the particularly optimal poses are chosen, ATR is slightly more robust than UTR. The fact that the Ankle Torque Reduction Strategy provides the best overall robustness gain is especially useful in the context that its optimal poses are the same as the robustness optimizing poses for normal LQR controllers. In contrast, the UTR strategy requires a more crouched steady-state pose in order to reach the same robustness.

So far, we have seen that different poses provide different levels of robustness. In Fig. 6, it is shown how large the smaller (in magnitude) of both positive and negative impulses can be, in dependency of the required steady-state effort for different poses and the applied control strategy. It can be seen that there is a trade-off between the steady-state balancing effort and the achievable robustness when choosing a robot pose. As expected, the steady-state effort minimizing pose is energy efficient but provides the lowest robustness margins. However, the robustness can be enlarged significantly by increasing the steady-state effort a little bit and switching to the ATR-optimized pose. This corresponds to leaning the humanoid a little bit forward, such that the overall CoM is close over the middle of the foot in x. The overall trend indicates that crouched poses with higher steady-state



(a) Joint input torques for ATR and UTR strategies over time. The discontinuities in slope of the torque plots within the first second indicate the switching between the joint torque reduction and the default control strategies. Afterwards, the joint torques converge to their steady-state values.



(b) The Center of Pressure ξ_x and the x-coordinate of the CoM (x_{CoM}) over time. Both the ATR and the UTR strategy place ξ_x at the edge of its Region of Trust at 0.18m. Note that the UTR strategy switches back about 0.05 seconds earlier to the default control law than the ATR strategy in this particular example. Afterwards, both x_{CoM} and ξ_x converge to their steady-state values.

Fig. 5: Example ATR und UTR strategy simulation results for a perturbation of $45\frac{\text{kgm}}{\text{sec}}$ and $y_{\text{CoM}} = 0.76$ m.

effort should be chosen in order to maximize robustness towards unexpected perturbations. Then, taking the pose which is optimized for ATR robustness clearly offers the best compromise between balancing effort and the minimal provided robustness margin.

C. An example which saves energy and gains robustness using ATR on a joint-locked robot.

Consider a robot standing in place and assume that it is able to lock its knee and hip joints mechanically, for example, using a magnetic clutch. Assume that the work required to lock the joints is small in comparison to the perturbation recovery efforts. With joints being locked, it can be represented as a one-link inverted pendulum model. One policy to pick an appropriate way of counteracting a perturbation is to choose the least costly way in terms of control effort. For example, one would typically choose an ankle strategy only, if the perturbation is sufficiently small. Increasing the magnitude of the impulse, one would begin to use additional degrees of freedom and unlock joints in order to recover, at the expense of consuming more energy. Fig. 7a shows the perturbation bounds for the joint-locked and unlocked cases with the poses being chosen subject to steady state effort minimization and using an LQR controller

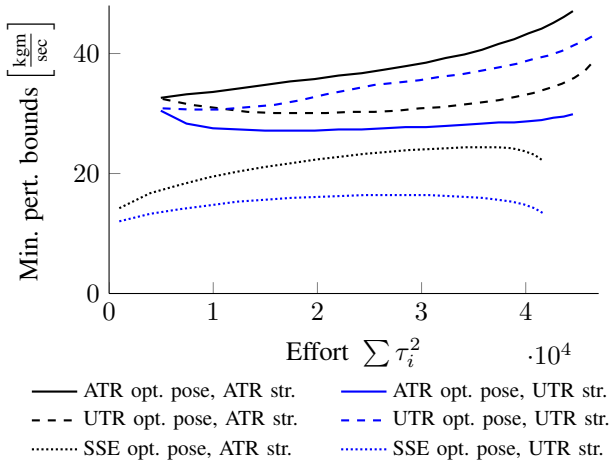
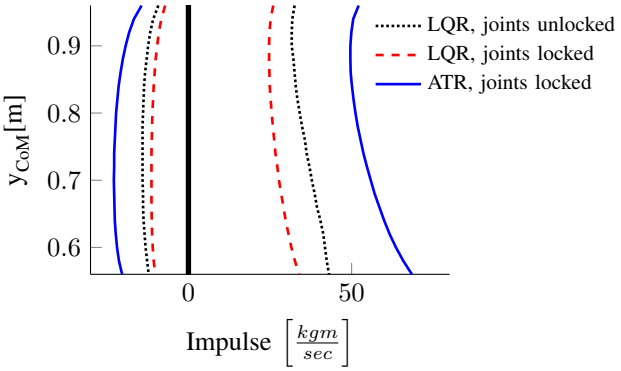


Fig. 6: Minimal admissible perturbation magnitudes of both positive and negative impulses for the ATR and UTR strategies (str.), plotted over the required steady-state effort for ATR-, UTR robustness optimizing and steady-state effort (SSE) minimizing poses.

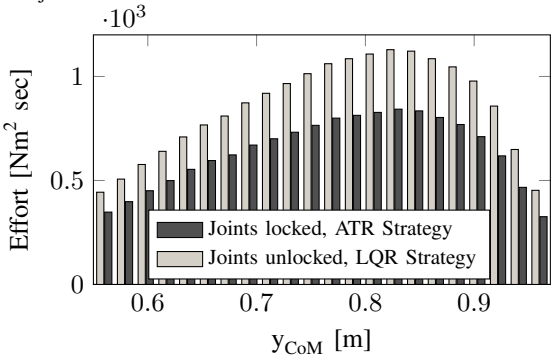
only. Furthermore, it shows the perturbation bound of a corresponding joint-locked robot using the ATR strategy. In the presented example, the joint-locked ATR-controlled system can take impulses of at least 46% larger magnitude than an LQR controlled system with all degrees of freedom being unlocked. At the same time, as shown in Fig. 7b, the effort measured in terms of an integral of squared input torques over the time needed for recovery is significantly smaller for the ATR joint-locked robot than for the LQR joint-unlocked case. That means, in this particular example, a joint-locked robot with ATR outperforms a fully actuated LQR controlled robot both in terms of robustness and recovery effort.

V. CONCLUSIONS

The purpose of this paper is to introduce a new approach of adapting joint torques for increased robustness of humanoid standing balance, and many advantages can be stated. First, we are now in a position to compute a set of input torques such that the CoP is placed at a desired location, based on a nonlinear model, with comparably low computational effort. Second, the presented strategies enlarge the robustness towards impulsive perturbations without having to invoke higher level strategies like taking a step or using an additional support. This results in the potential of a bigger variety of tasks which can be performed by a humanoid robot while balancing. The approach is general for n-link inverted pendulum models in the sagittal plane. Note that there is no need for an explicit trajectory planning of the CoP with the presented simple switching technique, which includes only one decision stage. The ATR and UTR adaption strategies themselves are not independent but serve as a robustness improving extension for a default feedback control law. A further advantage is that in order to maximize robustness for unexpected perturbations from both sides, it is not generally necessary to change the system's setpoint. It turned out that the optimal poses w.r.t. our robustness cost function are the



(a) perturbation bounds for LQR with unlocked joints and LQR/ATR for locked joints



(b) The ATR strategy with joints being locked is more energy-efficient in recovering from a perturbation than LQR with unlocked joints. In this example, the control efforts for the recovery from positive impulses which are equal to the perturbation bounds for the joint-unlocked, LQR-controlled case, are shown.

Fig. 7: A one-link model (locked knee and hip joints) using the ATR strategy outperforms a fully actuated three-link model with unlocked joints and LQR controller.

same for LQR and ATR, within our numerical resolution. A third and final point is that we can quantify the trade off between control effort at steady state and robustness to perturbations. In this context, it is possible to match a humanoid robot's standing pose (upright versus crouched) to particular operating conditions.

VI. DISCUSSION AND FUTURE WORK

Many topics can be identified for future work. First, no joint constraints were included for the actual process of recovering from a disturbance, however, this is not a major issue for crouched poses because the closed-loop does not overshoot during counteracting perturbations. No joint torque limits were taken into account and the most extreme ankle torque magnitude being recorded during all simulations was close to 200 Nm.

An extension of the presented approach to a more general class of perturbations, such as constant pushes or distributed forces, is also desirable. We only analyze a particular class of impulsive perturbations and we rely on numeric simulations: we give no analytic statements about the Basins of Attraction (BoAs), stability or speed of convergence (etc.) of the presented ATR and UTR strategies. For a real technical system, one would therefore still have to rely on pre-computed perturbation bounds and look-up tables to know when an initial condition is within the BoA for successful recovery. In practice, we anticipate that one would not have to cover the entire state space but only certain types and magnitudes of perturbations which would keep the look-up table dimensions small.

Additionally, considering using a robot model with two legs, an even bigger gain in robustness should be reachable by placing one foot a step forward and thus enlarging the overall base of support. The presented strategies are particularly helpful in scenarios with limited accessible ground area, for example for balancing on a small support region. A disadvantage of the switching algorithm presented in Fig. 3 is that an estimation of the CoP location using the default strategy is required - even if a joint torque reduction strategy is active. Using a model with an asymmetric foot, simulations have shown that for a small base of support, the strategy of adapting only the ankle torque (ATR) outperforms the strategy of reducing all joint torques uniformly. In contrast, UTR results in a bigger robustness margin than ATR if the base of support is rather large. Consequently, there must be a relation between the footsize and the best choice of a joint adaption strategy. The assumption about which direction the model is facing might also play an important role in that question. In this work, the Region of Trust was chosen arbitrarily. The choice should be optimized in future work.

Finally, investigations are necessary for what happens if the plant shows significant non-minimum phase behavior: the current switching strategy includes the risk of provoking closed-loop chattering if the absolute value of the CoP starts increasing again after a successful back-switch to the default controller. Although the simple approach presented in this paper worked well, a more sophisticated switching technique could be developed in future.

REFERENCES

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

[2] D. Winter, *A.B.C. (Anatomy, Biomechanics, Control) of Balance During Standing and Wa Lking*. Waterloo Biomechanics, 1995.

[3] L. M. Nashner and G. McCollum, "The organization of human postural movements: A formal basis and experimental synthesis," *Behavioral and Brain Sciences*, vol. 8, pp. 135–150, 2 1985.

[4] H. Hemami, K. Barin, and Y. Pai, "Quantitative analysis of the ankle strategy under translational platform disturbance.," *IEEE Trans on Neural Sys and Rehab Eng*, vol. 14, pp. 470–80, Dec. 2006.

[5] A. D. Kuo, "An optimal control model for analyzing human postural balance.," *IEEE Trans on Bio Eng*, vol. 42, no. 1, pp. 87–101, 1995.

[6] C. Liu and C. G. Atkeson, "Standing balance control using a trajectory library," *IEEE/RSJ IROS*, pp. 3031–3036, 2009.

[7] B. Stephens, *Push Recovery Control for Force-Controlled Humanoid Robots*. PhD thesis, CMU, Aug 2011.

[8] C. G. Atkeson and B. J. Stephens, "Gain scheduled control of perturbed standing balance," *IEEE/RSJ IROS*, pp. 4063–4068, 2010.

[9] H. Ono, T. Sato, and K. Ohnishi, "Balance recovery of ankle strategy: Using knee joint for biped robot," *2011 1st International Symposium on Access Spaces (ISAS)*, pp. 236–241, June 2011.

[10] Open Source Robotics Foundation, "DARPA Robotics Challenge - Gazebo Wiki," March 2013. http://gazebosim.org/wiki/DRC/.

[11] E. Otten, "Balancing on a narrow ridge: biomechanics and control," *Phil Trans of the Royal Soc: B, Bio Sci*, vol. 354, no. 1385, pp. 869–75, 1999.

[12] The MathWorks, Inc., "MATLAB Global Optimization Toolbox." http://www.mathworks.com/products/global-optimization/.