Dynamic Lateral Stability for an Energy Efficient Gait

Zhenglong Sun [a] Nico Roos [a]

[a] *Department of Knowledge Engineering, Maastricht University*
P.O. Box 616, 6200 MD Maastricht, The Netherlands

Abstract

This paper presents an energy efficient dynamically stable gait for a Nao humanoid robot. In previous work we identified a dynamically stable and energy efficient gait in the sagittal or walking direction of a Nao robot. This gait proved to be more energy efficient than the standard gait, provided by the manufacturer. Dynamic stability in the lateral direction was not addressed. Lateral stability was handled by full stiffness of the joint in lateral direction. In this paper we report on adding dynamic lateral stability. We do not yet incorporate feedback of sensors. This implies that the gait is only suited for flat horizontal surfaces that some lateral joint stiffness is needed in the implementation on the Nao.

1 Introduction

The gait of humans is often assumed to be the most energy-efficient way of walking [2]. Srinivasan and Ruina [10] confirm this hypothesis using a simple model in which the human is a point mass with straight legs that can change in length during a step. Their results show that the dynamically stable inverted pendulum walk is the most energy-efficient gait [7]. In previous work we demonstrated this also holds for humanoid robots such as a Nao, despite differences with humans. For instance, humans do not need to bend the knee of the stance leg while walking, because they can push off using the foot and the calf muscle. A humanoid robot such as a Nao, cannot push off using its foot. Instead it must provide the energy for maintaining a gait by bending and stretching its knee joint. Experiments with a Nao showed that the torque of knee joint as a result of bending the knee joint is the main source of energy consumption of the Nao during walking. A dynamically stable gait in the sagittal direction that minimize the knee bending proved to be the most energy-efficient gait.

We extend the dynamically stable gait in the sagittal direction with dynamic stability in the lateral direction. Simulations with an inverted pendulum model for the Nao with dynamic lateral stability showed that adding dynamic lateral stability does not change the dynamically stable gait in sagittal direction. We determined the requirements for the lateral stability and we adapted our gait controller incorporating dynamic lateral stability. The adapted gait controller has 8 control parameters, for which we learned the optimal values for a Nao through policy gradient algorithm. Figure 1 shows a high-level outline of the approach.

In the paper, we first briefly describe our previous work [11, 12] about a walking pattern generation which identifies a most energy-efficient gait without considering lateral stability (Section 2). The main contribution of this paper is to propose a method for lateral stability improvement (Section 3). The results are used to create a gait controller (Section 4) which is fine-tuned for a Nao robot using a *webots* simulator (Section 5). The gait is evaluated on a real Nao robot (Section 6). The last section concludes the paper.

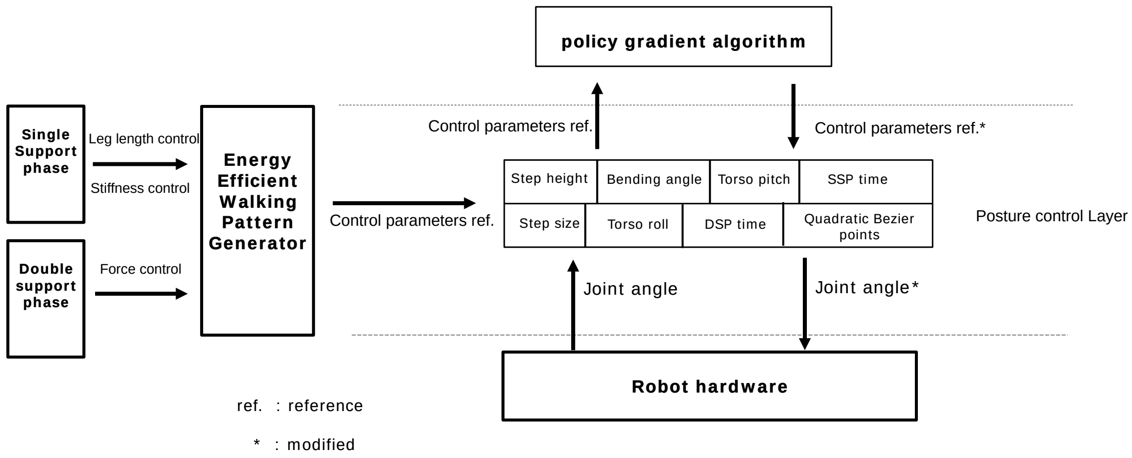

Figure 1: Outline of the proposed approach, showing the details about work flow

2 Walking Pattern Generation

The gait of humans and of humanoid (bipedal) robots is a repeating pattern consisting of two phases; a single support phase (SSP) where the body is supported by only one leg and a double support phase (DSP) where the body is supported by both legs [5]. In the DSP the weight of the body is shifted from one leg to the other. The DSP is crucial for the lateral stability but is sometimes ignored when analysing the gait. However, since it is impossible to implement a gait on a Nao without a double support phase, we must consider it in our model. We will start presenting a model without a DSP and subsequently extend the model with a DSP.

2.1 Single Support Phase

To analyse the energy consumption, we developed an Inverted Pendulum model with telescopic legs [11, 12]. This model, which is based on the work of Srinivasan and Ruina [10], allows the length of the support leg to vary during a step. A leg-length policy $\delta : [-\frac{\pi}{2}, \frac{\pi}{2}] \to [0, 1]$ determines how much the stance leg will be shortened as function of the angle β between stance leg with vertical axis. The shortening of the stance leg is realized by bending the knee joint. Experiment with a Nao robot showed the the torque on the knee joint is the main factor determining the energy consumption during walking and that the energy needed to stretch the stance leg during a step can be ignored. Using simulations in Matlab, an optimal leg-length policy that minimize the energy consumption, was determined [11, 12]. The optimal leg-length policy shows that the SSP starts with a slightly bended stance leg which is subsequently stretched. After stretching the stance leg remains stretched till the end of the step.

2.2 Double Support Phase

We extended the model described in the previous subsection with double support phase [4]. The length of the SSP during a step will be a parameter of our controller.

We need a way to describe the influence of the swing leg on the mass in DSP. This can not be done by just simply applying a leg-length policy for the swing leg in the double support phase. Given the step size s, the leg-length policy $\delta(\beta)$ of the stance leg and angle β, the length leg of the swing leg is fixed. Prescribing the length of the swing leg by a policy, creates a rigid triangle in which the mass m can no longer move freely. We therefore choose to let the mass m move freely given the leg-length policy of the stance leg and use a force policy for the swing leg in the double support phase. So the length of the swing leg is determined by the leg-length policy of the stance leg, the step size and the angle of the stance leg, but the force that swing leg executes on the mass is determined by the force policy of the swing leg. This force may influence the sagittal speed of the mass m.

After adding the double support phase and the force policy for the swing leg during the double support phase, we re-run our Matlab simulation. The result of this simulation showed that the optimal

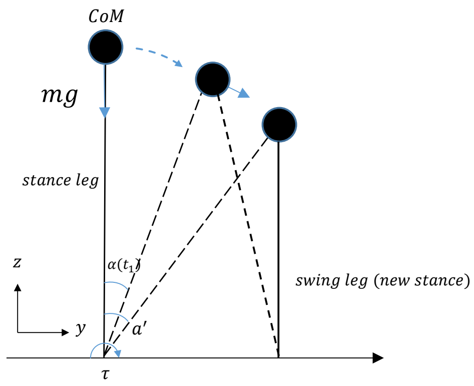

Figure 2: The lateral plane during double support phase

leg-length policy did not change and the optimal force policy is to put no force on the swing leg till it becomes the new stance leg. We also evaluated the effects of different force polices on leg-length policy. The Matlab experiments showed that the shape of the leg-length policy does not change. The robot still starts with a slightly bended leg which is subsequently stretched and remains stretched till the end of the step. During the DSP, the stance leg is always stretched.

In the double support phase the robot has to shift its weight from the stance leg to the swing leg. In order to keep balance in lateral direction at the end of the double support phase, the robot must put force on the swing leg to stop the lateral movement in time. Adding this observation to our model and to our gait controller for the Nao robot is the main contribution of this paper.

3 Lateral Control

To improve the stability of the walking pattern generation described previously, we exploit lateral controller to regulate the CoM lateral movement and velocity during double support phase. Our idea is to use the force generated by the swing leg and and upper body tilt to regulate lateral component of CoM velocity. Since this work focus on the lateral component of the walking motion, and our experiments with different leg-length polices showed the stance leg is always stretched during the DSP, we restrict the equations to the lateral plane. Missura and Behnke [8] confirmed that sagittal and lateral controllers can be modeled independently.

In order to introduce the equations describe the movement of CoM in lateral plane, we first define a variable $\alpha(t)$ which is the angle between the stance leg and the vertical axis. When the total force resulting from gravity and inertia generates rotation around the contact point between the sole of stance leg and ground, the angle $\alpha(t)$ varies from 0 to α', as illustrated in Figure 2.

We assume that during the single support phase, the robot is perfectly balanced in the lateral direction. Therefore, at the beginning of DSP, the stance leg is vertical to the ground ($\alpha(0) = 0$), and in the lateral plane there is no torque making the CoM rotate around the sole of the stance leg. In order to generate the torque τ rotating the CoM from $\alpha(0)$ to α', we manipulate the upper body to bend slightly inwards at angle ω. The bending ω disrupts the balance enabling gravity to create a torque $\tau > 0$. We manipulate the force generated by the swing leg to control the rotation of the CoM with a non-zero angular velocity $\dot{\alpha}$ and to stop at the position ($\alpha(t) = \alpha'$) where the robot can put its whole body weight on the new stance leg and keep it stable. The problem is to control the torque τ appropriately. Our method to mitigate the problem is controlling the force generated by the swing leg by means of a force policy. Compared to the height of CoM, the step size is small (less than 5% of CoM height). Therefore we assume that in the DSP, the length of stance leg can be considered as fixed in the lateral plane. That is, on the stance leg, we ignore the effect of the CoM moving forward in the sagittal plane. Given this assumption, we analyse the forces on the CoM in the lateral plane. We break down the gravity into two components consisting of a radial force F_r along the stance leg and F_p that is perpendicular to the stance leg (see Figure 3.a). We manipulate the force policy $\gamma(\alpha)$ of the swing leg to control the force F_q generated by the swing leg on the CoM. We also break down F_q into two components: F_b which directs opposite of F_r and another force F_p' which is opposite to F_p (see Figure 3.b). Therefore, in the radial direction, there is the combine radial force $F_n = F_r + F_b$. Since the length of stance leg l is fixed, the radial force F_n has no effect on the movement of the CoM. However, the combine force $F_t = F_p + F_p'$ generates a torque τ around the contact point between the sole of stance leg and ground. The torque τ rotates the CoM, shifting the weight from stance leg to the swing leg.

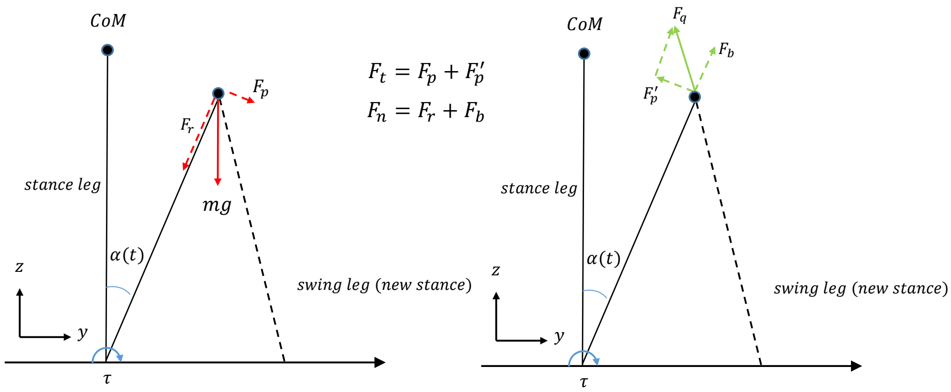

Figure 3: (a) Force on stance leg in lateral plane and (b) Forces on swing leg in lateral plane

At the beginning of DSP, ideally, the force policy $\gamma(\alpha)$ imposes no force on swing leg. Therefore the torque τ generated by the slightly inward bending ω of upper body is needed to start the lateral movement of the CoM. As $\alpha(t)$ increases, the force policy controls the force F_q to gradually decrease the $\dot{\alpha}(t)$, and stops the CoM movement when $\alpha(t) = \alpha'$. The force F_q causes the force F_p' which de-accelerates the movement of the CoM. When $\alpha(t) = \alpha'$, the force $F_t = 0$ and the torque $\tau = F_t \cdot l = 0$, and therefore the CoM will stop rotating. We generate a force policy by regulating the knee stiffness of the swing leg, as a function of the angle $\alpha(t)$. The shape of the force policy is determined by means of Quadratic Bezier curves, as illustrated in Figure 4. The Quadratic Bezier curves is defined by 3 points in the interval of the DSP. The start point and the end point are fixed, so we start with no force generated by the swing leg and stop with the full weight of the robot on the swing leg, after which it becomes the new stance leg. We assume a smooth transition between these two points which is determined by the middle point ζ of the Quadratic Bezier curves. So we have to determine the optimal point ζ.

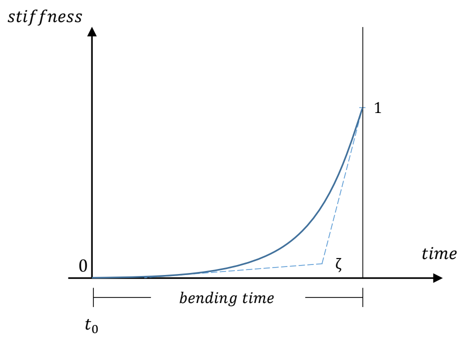

Figure 4: Stiffness over time by Quadratic Bezier Curves

To summarize, the controller manipulates the upper body to bend slightly inwards at angle ω to trigger the CoM movement. At first there is no stiffness on swing leg, therefore the magnitude of F_p' is 0. Consequently, F_t leads to an acceleration of the angular velocity of CoM around the sole of the stance leg, which make the knee joint of the swing leg start to bend. Next, the stiffness of the keen joint on swing leg is increased in order to stop the rotation when the CoM reaches its end position ($\alpha(t) = \alpha'$).

4 Optimizing Gait Parameter for Nao

This section describes the learning of the optimal control parameter of a dynamic gait for a Nao.

4.1 Gait Parameters

This section presents the parameters of a gait that realizes the leg-length policy determined by the experiments described in the previous section. Based on the results of the simulation experiments, we identify 8 parameters (the new parameters to this work are the Quadratic Bezier points and the Torso Roll inclination, see Section 3) that are essential in controlling a dynamic gait:

- *Step Size* (θ_1): Defines the how long Nao can move in a singe step (sagittal).
- *Step Height* (θ_2): Defines the maximum distance between ground and lifting feet. A high step height will require higher speed of the swing leg and may cause horizontal instability. A low step height increases the possibility of tripping and limits the step size.
- *Knee Bending* (θ_3): Defines the maximum bending of the swing leg at the beginning of the double support phase. This parameter determines the sagittal velocity and the energy cost.
- *SSP Time* (θ_4): Defines how long the single support phase lasts. This parameter determines the sagittal walking velocity.
- *DSP Time* (θ_5): Defines how long the double support phase lasts. This parameter determines the duration of the swing leg (the next stance leg) compression to θ_3 in the double support phase.
- *Torso Pitch Inclination* (θ_6): Defines the maximum angle that torso leans in sagittal direction. If positive, it will move the center of mass (CoM) in sagittal direction. If it is set not appropriate, a fall will occur.
- *Quadratic Bezier points* (ζ): Defines the magnitude of middle points in Quadratic Bezier Curves which determines the ground reaction force on swing leg (introduced in Section 3).
- *Torso Roll Inclination* (ω): Defines the maximum angle that torso leans in lateral direction. If positive, it will move the center of mass (CoM) towards the swing leg in frontal view. If it is set not appropriate, instability will occur (introduced in Section 3).

Table 1: Trajectory Parameters in Sagittal Plane

Description of joint motion	q
step size	θ_1
swing hip pitch	$p_1(\theta_1, \theta_2, \theta_4, \theta_6)$
swing knee pitch	$p_2(\theta_1, \theta_2, \theta_4, \theta_6)$
swing ankle pitch	$-p_1 - p_2$
stance hip pitch	$p_3(\theta_1, \theta_2, \theta_3, \theta_5, \theta_6)$
stance knee pitch	$p_4(\theta_1, \theta_2, \theta_3, \theta_5, \theta_6)$
stance ankle pitch	$-p_2 - p_4$

Table 1 shows all the parameters of the trajectory for walk movement. The walk posture q is determined by joints value, step size, acceleration and so on. The value of hip pitch, knee pitch and ankle pitch are functions p_n of parameter sets.

4.2 Policy Gradient Algorithm

After investigating several policy search algorithms [3, 9], we chose to use a policy gradient method presented by Kohl and Stone [6] to optimize the Nao's gait. This method is among the Finite-different methods and quite straightforward to understand. In the control law optimization experiment [9], the finite-different methods turned out to be less efficient than other prominent general approaches and converge to local optimal in motor planning experiments, Nevertheless, if we cannot differentiate the polices w.r.t the control parameters, the finite different methods becomes the only option applicable [9]. In this method, the objective function *Q* is a function to be optimized for the energy cost and stability.

The policy gradient method starts with an initial parameter vector $\pi = \theta_1, ..., \theta_N$ and estimates the partial derivative of the objective function *Q* with respect to each parameter. This is done by evaluating

t randomly generated policies $R_1, ...R_t$ near π, such that each $R_i = \theta_1 + \delta_1, ..., \theta_N + \delta_N$ and δ_j is randomly chosen to be either $-\epsilon$, 0, $+\epsilon$, where ϵ is a small fixed value relative to θ. After evaluating each policy R_i on the objective function Q, each dimension of every R_i is grouped into one of the three categories to estimate an average gradient for each dimension:

$$R_i \in \begin{cases} S_{+\epsilon,n}, & \text{if the } n\text{th parameter of } R_i \text{ is } \theta_n + \epsilon_n \\ S_{+0,n}, & \text{if the } n\text{th parameter of } R_i \text{ is } \theta_n + 0 \\ S_{-\epsilon,n}, & \text{if the } n\text{th parameter of } R_i \text{ is } \theta_n - \epsilon_n \end{cases}$$

We calculate average score $Avg_{-\epsilon,n}$, $Avg_{+0,n}$ and $Avg_{+\epsilon,n}$ for $S_{-\epsilon,n}$, $S_{+0,n}$ and $S_{+\epsilon,n}$ respectively.

- $Avg_{-\epsilon,n}$ average score for all R_i that have a negative perturbation in dimension *n*
- $Avg_{+0,n}$ average score for all R_i that have a zero perturbation in dimension *n*
- $Avg_{+\epsilon,n}$ average score for all R_i that have a positive perturbation in dimension *n*

These three average values estimate the benefit of altering the nth parameter by $+\epsilon_n$, 0, $-\epsilon_n$. An adjustment vector A of size n is calculated where

$$A_n = \begin{cases} 0, & \text{if } Avg_{+0,n} \geq Avg_{+\epsilon,n} \text{ and} \\ & Avg_{+0,n} \geq Avg_{-\epsilon,n} \\ Avg_{+\epsilon,n} - Avg_{-\epsilon,n}, & \text{otherwise} \end{cases}$$

In order to generate a gait that is energy efficient and stable, we adopt an objective function based on

Algorithm 1 Pseudo-code of Policy Gradient Algorithm

```
π ← InitialPolicy
while !done do
    R_1, R_2, ...R_t = t random perturbations of π
    evaluate(R_1, R_2, ...R_t)
    for n = 1 to N do
        Avg_{+ε,n}
        Avg_{+0,n}
        Avg_{-ε,n}
        if Avg_{+0,n} > Avg_{+ε,n} and Avg_{+0,n} > Avg_{-ε,n} then
            A_n ← 0
        else
            A_n ← Avg_{+ε,n} - Avg_{-ε,n}
        end if
    end for
    A ← A/|A| * η
    π ← π + A
end while
```

the energy cost and the stability. The energy cost is expressed by the normalized current M_c, and the stability by normalized standard deviation of the three accelerometers M_a

$$Q = 1 - (w_c M_c + w_a M_a) \tag{1}$$

The components of the objective function are weighted by w_c and w_a respectively to optimize for desirable goal. These weights are constrained so that the sum of the weights are equal to one. In this experiment, we set w_c = 0.25, w_a = 0.75.

4.3 Learning optimal parameters in simulator

To generate the optimal gait parameters and test the gait's performance, two separate experiments were conducted. Firstly, we upload local modules into simulator *Webots* to run the policy gradient algorithm.

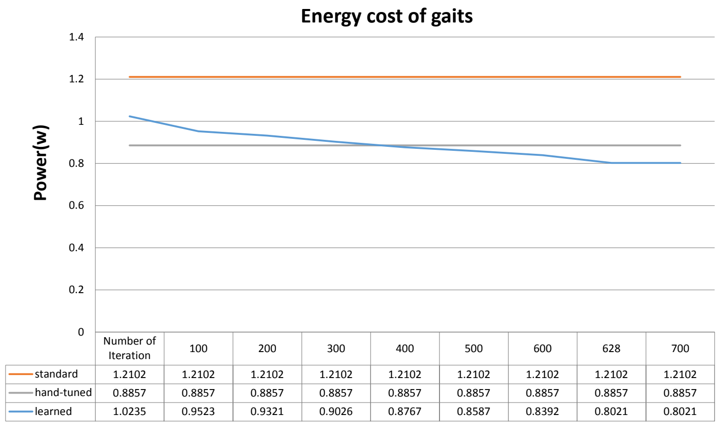

Figure 5: Comparison of Energy Cost of Three Gaits

We used a relatively elementary hand-tune gait as a starting policy for the policy gradient algorithm described in Section V. Though a bad starting policy may lead to a simulation failure, we did not deliberately optimize the starting policy. The performance of the task is measured by monitoring the knee torque and electric current. Falling or "toddle" is penalized. We found a most energy-efficient and dynamically stable gait after 628 iterations. Subsequent evaluations showed no further improvement. The learning algorithm produce a set parameters of stable gait which is more energy efficient and faster than the standard walk of Nao. The parameters of the gait and ϵ value are given in Table 2.

Table 2: Initial Parameters and Best Learned Parameters

Parameter	Initial Value	ϵ	Learned Value
step size	6(cm)		6(cm)
step height	3(cm)	0.02	3.24(cm)
knee bending	15 (degree)	0.1	13.8 (degree)
SSP time	300(ms)	25	225(ms)
DSP time	300(ms)	25	375(ms)
torso pitch inclination	10 (degree)	0.1	7.5 (degree)
torso roll inclination	10 (degree)	0.1	6.8
Quadratic Bezier points	(0.5*DSP time, 0.5)	0.1	(0.9*DSP time, 0.2)

5 Simulation and Real World Evaluation

To validate our approach, we perform a real world experiment with the Nao humanoid robot, which has 25 degrees of freedom. We validate the result of the learned parameters by sending them to a Nao robot and command it to walk a constant distance. We compared the energy consumption of learned gait with the energy consumption of the standard gait of the Nao and our initial hand-tuned gait. The step size was set to 6 cm. Figure 5 shows the energy consumption of the three gaits. We see that the learned gait results in a power reduction of 33.7% of standard gait and a reduction of 9.4% of hand-tune gait. The accompanying video material[1] shows the Nao robot walking on flat ground with our proposed gait controller.

6 Conclusion

In this paper, we presented a framework to generate energy efficient dynamic human-like walk for a Nao humanoid robot. Based on previous work, we proposed a simple lateral control method for a gait with dynamic lateral stability. We optimize the control policy for a Nao humanoid robot and evaluated the

[1]https://project.dke.maastrichtuniversity.nl/robotlab/wp-content/uploads/naowalk.mp4

result on the real Nao. The result shows the gait we proposed is more energy efficient and dynamically stable than the standard Nao gait.

References

[1] JJ Alcaraz-Jiménez, D Herrero-Pérez, and H Martınez-Barberá. A closed-loop dribbling gait for the standard platform league. In *Workshop on Humanoid Soccer Robots of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2011.

[2] Steven H Collins and Andy Ruina. A bipedal walking robot with efficient and human-like gait. In *ICRA*, pages 1983–1988. IEEE, 2005.

[3] Marc Peter Deisenroth, Roberto Calandra, André Seyfarth, and Jan Peters. Toward fast policy search for learning legged locomotion. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1787–1792. IEEE, 2012.

[4] Joel A DeLisa. *Gait analysis in the science of rehabilitation*, volume 2. Diane Publishing, 1998.

[5] Arnaud Hamon, Yannick Aoustin, et al. Optimal walking gait with double support, simple support and impact for a bipedal robot equipped of four-bar knees. *IMSD 2012*, 2012.

[6] Nate Kohl and Peter Stone. Machine learning for fast quadrupedal locomotion. In *AAAI*, volume 4, pages 611–616, 2004.

[7] Arthur D Kuo. The six determinants of gait and the inverted pendulum analogy: A dynamic walking perspective. *Human movement science*, 26(4):617–656, 2007.

[8] Marcell Missura and Sven Behnke. Lateral capture steps for bipedal walking. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 401–408. IEEE, 2011.

[9] Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2219–2225. IEEE, 2006.

[10] Manoj Srinivasan and Andy Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439:72–75, 2006.

[11] Zhenglong Sun and Nico Roos. An energy efficient gait for a nao robot. *BNAIC*, 2013.

[12] Zhenglong Sun and Nico Roos. An energy efficient dynamic gait for a nao robot. In *IEEE Conference on Autonomous Robot Systems and Competitions*, 2014.