RL-Policy Guided Optimal Design of Parallel Elastic Actuator for Weak Actuation of Bipedal Robot

Alfred Cueva*
*Department of Mechanical Engineering, Seoul National University, Seoul, Korea

***Abstract*—The use of parallel elastic actuators provides additional torques in bipedal robots. However due to existing constraints like available power or joint's range of motion, optimal actuators should be carefully designed. Existing solutions involve modifying the actuator system and often rely on intuition, making efficient testing challenging. In this work we introduce a framework to systematically optimize actuator design of parallel elastic actuators for bipedal robots. We design additional virtual stiffness to joints and optimize its parameters based on a cost function involving maximum torque and energy. We first train a policy with model-free Reinforcement Learning using Potential Based Rewards (PBRS) encouraging symmetry and adequate torque consumption. From the time series data from the policy roll out we find the optimal actuator parameters using Bayesian Optimization. With this method we obtain optimal parameters to design parallel elastic actuators that reduce the torques needed for the 6-dof Bolt Robot. The optimal actuators are tested under various criteria. The experiments show that we can increase the maximum forward velocity of the robot by 19% compared to our baseline while encouraging velocity and torque tracking.**

I. Introduction

In recent years bipedal robots has revolutionized the field of robotics but it still faces challenges when it comes to energy consumption. To this end, researchers have utilized used Reinforcement Learning (RL) methods to learn bipedal locomotion policies while considering energy expenditure as a key objective. One of the early researches [1] showed that energy minimization plays a crucial role to achieve natural gait patterns in quadrupedal legged robot. The approach used a distillation-based learning pipeline with a velocity-conditioned policy and rewards penalizing energy. Another approach [2] used evolving policy parameterization and passive compliance to find optimal CoM trajectories and minimize energy consumption in the COMAN Robot.

However, the various physical interactions of legged robots with the environment rather calls for actuator designs that aim to maximize torque, bandwidth, and power while minimizing losses from friction, inertia, and mass [3]. Implementing powerful actuators has shown great promise for legged robots as they provide additional torques [4]. A promising technique is the incorporation of Parallel Elastic Actuators (PEA), where the spring and actuator are in parallel, in the robot's leg design. These elements act as passive mechanical springs, storing and releasing energy during locomotion. The robot STEPPR [5], a bipedal robot using parallel-springs at the ankle and hip is a notable example. By effectively recycling energy, the parallel elastic elements reduced the overall energy expenditure in 13% required for walking, leading to improved energy efficiency and extended operation times. Some researchers [6] took this approach further and designed serial-parallel hybrid legs for humanoid robot reducing required torques. These previous works indeed help reducing leg inertia and encourage uniform gait but they require making various assumptions to simplify the control theory and meet physical constraints [7] calling for formal algorithmic methods to optimize actuator design.

As traditional methods used intuition, design optimization techniques have been developed. Bauer et. al. [8] proposed numerical optimization of elastic couplings and joint angle trajectories to minimize average energy consumption. Using linear torsion springs they increased energy efficiency by 50%. However this method is not suitable for more complex bipedal robots as it depends on the reference trajectory of two rigid kneeless legs.

More recent methods [9] introduced using a hardware policy along with a control policy in model-free RL to obtain optimized weights and define hardware parameters. It remains yet a challenge to incorporate mechanism kinematics and complex morphologies as computational graphs, specially for legged locomotion. A similar approach was proposed by Bjelonic et. al. [10] co-optimizing the design and controller in the ANYmal [11] quadrupedal robot. They use privileged learning [12] and Heteroscedastic Evolutionary Bayesian Optimisation (HEBO) algorithm [13] to optimize an objective function (tracking performance) as black-box optimization since these physical quantities aren't usually differentiable by design parameters.

Although they implemented parallel linear springs in the knee of the robot using RL-based control and surrogate models, it could be inaccurate to model the workspace of the knee joint with cable-spring mechanisms [10]. One major challenge still remains, to find optimization approaches generalizable to all joints of bipedal robots, enabling them to generate higher torques with less dependency for external power supply.

Therefore in this work we propose an optimization framework to find optimal parameters to design parallel elastic actuators (PEA) for weak actuation of bipedal robots. Our approach was tested under various criteria: Cost of transportation, Maximum Achievable velocity and tracking analysis. The major contributions of the proposed systematic approach for PEA design of legged robots are as follows:

- Optimizing the design parameters of Parallel Elastic

Actuators for all joints of a Legged Robot using model-free RL and Bayesian Optimization.
- An approach to obtain a locomotion control policy that consumes less motor power.
- Experimental results in simulation demonstrating the influence of the optimal actuators and added masses on the robot's motion.

II. PRELIMINARIES

A. Reinforcement Learning

To model the locomotion control problem, we use a policy π in model-free RL, building on Markov Decision Processes (MDP) [14]. In our 6-dof biepdal robot the policy π has an input of $x_t \in \mathbb{R}^{33}$ and previous action $a_{t-1} \in \mathbb{R}^6$, as each robot leg has three joints each. They are used of predict the current action a_t using Equation (1) and receive a reward r_t. The action a_t can be interpreted as positions per joint given to a PD controller or directly scaled as torques.

$$a_t = \pi(x, a_{t-1}) \tag{1}$$

RL methods aim to obtain an optimal policy π_* that can maximize the accumulated rewards discounted every time step from (2) using x_t and a_{t-1} to obtain a_t.

$$V = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right] \tag{2}$$

This iterative interaction with the environment yields $\tau = \{(x_0, a_0, r_0), (x_1, a_1, r_1), \ldots\}$ as the agent trajectory when using the locomotion policy π and transition probability density $p(\tau|\pi)$ for a given discount factor γ. Current methods parallelize this approach for thousands of robots [15] [16] [17] using Proximal Policy Optimization (PPO) [18] for policy optimization .

B. Bayesian Optimization with Gaussian Processes

Bayesian optimization (BO) [19] minimizes an unknown function $f : X \to \mathbb{R}$ within a limited budget of N function evaluations on a compact subset $X \subset \mathbb{R}^d$.

It employs a Gaussian process surrogate model $\text{GP}(x|\mu, \sigma^2, \theta)$ with hyperparameters θ estimated using Markov Chain Monte Carlo (MCMC) from m samples $\Theta = \{\theta_i\}_{i=1}^m$.

Given dataset $X = \{x_{1:n}\}$ and outcomes $y = \{y_{1:n}\}$ at step n, the model predicts a new query point x_q with i-th hyperparameter sample θ_i as $y_q \sim \frac{1}{m} \sum_{i=1}^m \mathcal{N}(\mu_i, \sigma_i^2|x_q)$ [20]. Being $k_i(x_q, X)$ the corresponding cross-correlation vector of the query point x_q with respect to the dataset X, the prediction involves:

$$\mu_i(x_q) = k_i(x_q, X) K_i(X, X)^{-1} y \tag{3}$$

$$\sigma_i^2(x_q) = k_i(x_q, x_q) - k_i(x_q, X) K_i(X, X)^{-1} k_i(X, x_q) \tag{4}$$

The decision process starts with an initial design of p points via Latin Hypercube Sampling (LHS) to mitigate bias. Subsequent points are chosen using the expected improvement criterion: $EI(x) = \sum_{i=1}^m [(\rho - \mu_i(x))\Phi(z_i) + \sigma_i(x)\phi(z_i)]$, where $\rho = y_{\text{best}}$. At iteration n, the next query point x_n is selected as $x_n = \text{argmax}_X EI(x)$.

III. METHOD

A locomotion policy is first learned using model-free RL and Potential Based Rewards to obtain time series data (joint position, joint velocity, joint torques). Then, we design a cost function which includes the maximum torque generated and energy. Using the obtained data and a cost function we find optimal parameters (stiffness and initial joint angle) through Bayesian Optimization for PEAs in all joints. Since the legs of a bipedal robot has 3 joints, 3 stiffness parameters k and 3 initial joint positions q_0 should be considered. Thus, the vector $\theta^* = [k^*_{\text{Hip Roll}}, k^*_{\text{Hip Pitch}}, k^*_{\text{Knee Pitch}}, q^*_{0\text{Hip Roll}}, q^*_{0\text{Hip Pitch}}, q^*_{0\text{Knee Pitch}}]^T \in \mathbb{R}^6$ represents the design space. The overall methodology is illustrated in Figure 1.

A. RL-Based Control

We trained our policy using a Deep RL framework [17], to train multiple environments simultaneously, and Potential Based Rewards (PBRS) [21] for robust humanoid locomotion, ease learning and encourage faster convergence to optimal solutions show in Appendix . The reward function includes metric to follow the linear base commands in x, y directions and rotational motion. We also punished torque sum to encourage realistic torque consumption for the robot with limited available torque or not. Similarly, to penalize unsymmetrical motions, we penalize the difference in joint position from hip roll and hip pitch using a helper squared exponential function as shown in (5) .

$$R_{\text{symmetry}} = \frac{1}{2 * \text{noise\_scale}} \times (\exp((q_1 - q_4) + (q_2 - q_5))) \tag{5}$$

The commands were x, y linear velocities and yaw while joint torques were the actions. The observations were position and velocity for base and joints, commands, contacts, and projected gravity for a total of size of 33 with their respective scales and no noise. All the training was done in simulation using 4096 environments for 1000 policy iterations and 100 Hz of control frequency. The simulator used was NVIDIA's Isaac Gym simulation environment which trains thousands of robots simultaneously using curriculum learning and Proximal Policy Optimization (PPO).

B. Design Objective

The goal of our approach is to find $\theta^* \in \mathbb{R}^6$, the design parameters for PEAs at joints, so that overall energy consumption of the bipedal robot is reduced. We roll out the designed locomotion policy in a flat terrain environment and no PEA added as baseline. Using the data from the simulation we utilize it in the cost function J (6) for later Bayesian Optimization. Since our main goal is to reduce the overall torque the joints need to produce, we define J as the sum of maximum torque produced and energy.

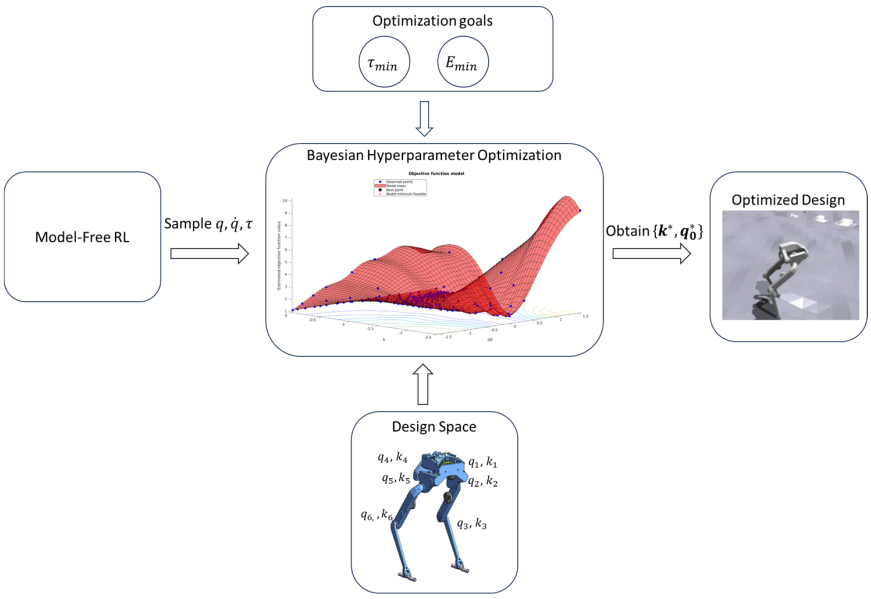


Fig. 1: Overview of the proposed design optimization approach. We first train a control policy using Model-Free RL to sample position, velocity, and torque of the robot. Then we perform Bayesian Optimization for a defined cost function and design space.

$$J = |\tau_{\max}| + \alpha \sum_{t} (\tau^T \dot{q}) \tag{6}$$

Since the Torque distribution from the RL policy roll-out has disturbances, $|\tau_{\max}|$ is the result of stochastic clipping the torque data. By doing so, we remove outliers and evaluate the maximum torque generated from our policy more accurately. Meanwhile, the energy term in J is the summation over all time steps t of the product of joint torque τ and joint velocity $\dot{q}$ from the policy roll-out. Because at heavier weights the joints are bound to produce higher powers we add a regularization term α towards prioritizing the effect of maximum torque generated when optimizing the cost.

On another hand, simulating the overall energy is difficult as there are various source of energy consumption like mechanical energy of actuators as well as transmission and electronics losses [10]. In this work we assume that the total energy of the robotic system can be estimated from the joule heating of individual actuators.

C. Optimization

With the defined cost function J we can find the optimal parameters $\theta^* \in \mathbb{R}^6$ for a specific task. The defined task was standard bipedal walking with appropriate velocity tracking. As our objective can't be directly differentiated by design parameters θ due to the discrete changes of foot contact during walking, we use Bayesian Optimization with Gaussian Processes as it uses a surrogate model and is gradient-free. The formalized optimal design parameters for PEA would be:

$$\theta^* = \operatorname{argmin} \mathbb{E}_{\theta \in \mathbb{R}^6}[J(\theta, \pi)] \tag{7}$$

Because we also aim for symmetry and naturalness of locomotion, that is directly enforced it by constraining the optimization variables. Namely, considering all joints for the bipedal robot from the design space we establish the following:

$$\text{Hip Roll:} \quad k_1 = k_4, \quad q_{0_1} = -q_{0_4} \tag{8}$$

$$\text{Hip Pitch:} \quad k_2 = k_5, \quad q_{0_2} = q_{0_5} \tag{9}$$

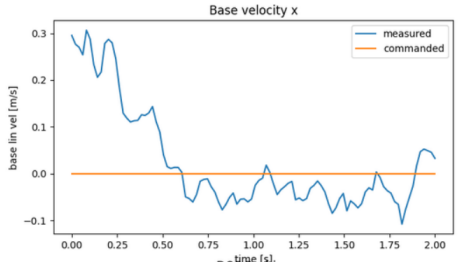
$$\text{Knee Pitch:} \quad k_3 = k_6, \quad q_{0_3} = q_{0_6} \tag{10}$$

IV. EXPERIMENTS

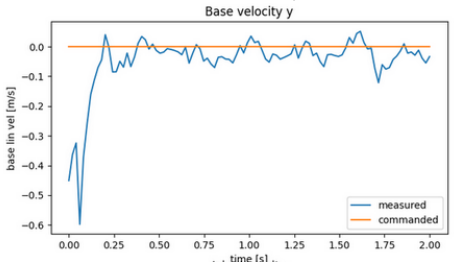
In this section we implement the PBRS rewards, obtain the optimal design parameters (stiffness and initial joint angle) from the policy roll-out and then test the optimized designs in motion under various metrics.

A. Control Policy

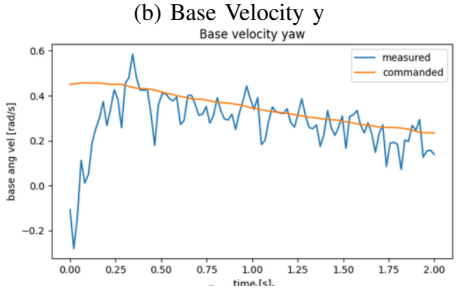
The results using the joint regularization reward from Equation 5, generate a more stable gait compared to traditional tracking penalties as shown in Fig. 2. The forward velocity (x direction) and the yaw velocity in the base of the robot both converges to the commanded velocity after 0.3 time step. Meanwhile, the sideways velocity converges faster, before 0.25s. However, all of the above measure velocities don't actually match the commanded values. Similarly, the torques



(a) Base Velocity x



(b) Base Velocity y



(c) Base Velocity yaw

Fig. 2: Baseline Velocities

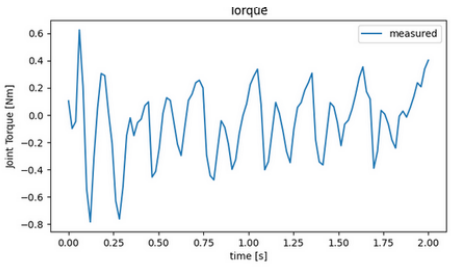
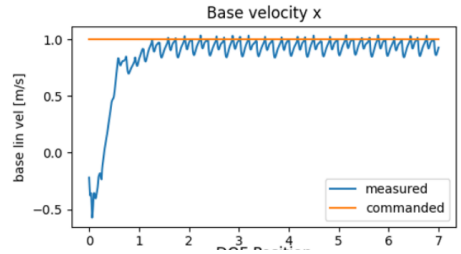


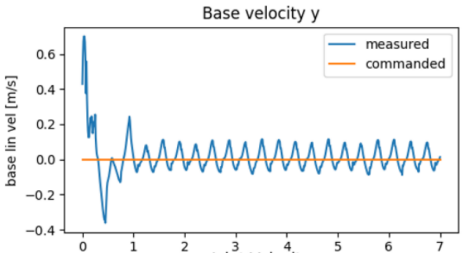
Fig. 3: Baseline Torque

also don't follow a uniform behaviour as seen in Figure 2 with a maximum achievable torque of 0.6.

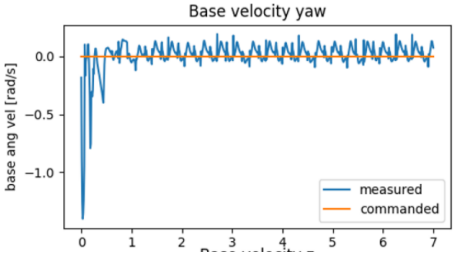
On the other side when implementing the potential based rewards for forward motion we see more consistent overall results. As seen in Figure 4, the robot fully converges to the commanded velocity after 1s. Although it took more than the baseline, the measure velocity is more consistent as clearly seen for Base Velocity y of Fig. 4. It takes time to reach the forward and sideways commanded velocities but for the yaw it happens instantaneously. This is because the potential based rewards in theory do encourage faster convergence compared to standard velocity/torque tracking rewards, specially if joint regularization is used. Moreover, as seen in Fig. 5 the new mean measured torque has a peak over 1 Nm if positive and well over 1.5 if negative, both bigger than the baseline torques.



(a) Base Velocity x



(b) Base Velocity y



(c) Base Velocity yaw

Fig. 4: PBRS Velocities

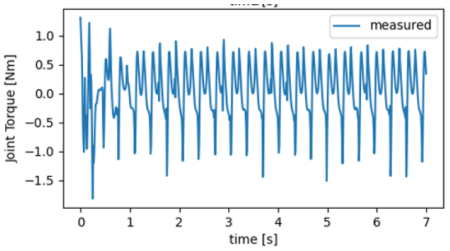


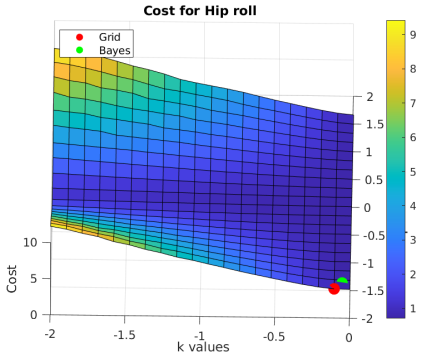
Fig. 5: PBRS Torque

B. Optimal Parameters

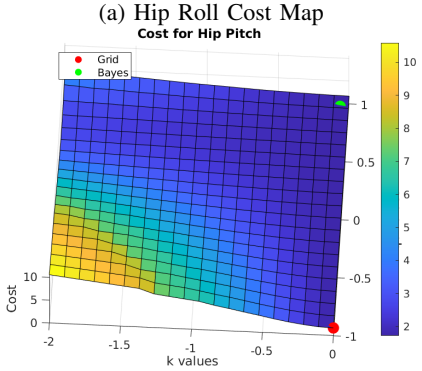
With better tracking from to PBRS rewards, we roll-out the policy and obtain optimal parameters $\theta^* \in \mathbb{R}^6$ for the robot using our Cost Function from Equation 6 constrained by 9, 8, 10 and Bayesian Optimization. Because PEAs are expected to provide additional torque, we obtain the policy data from the rollouts of when the mas of the base increases by 1kg, 2kg and 3kg.

Using our Cost Function we can build a 20x20 grid according to k and q values. We compare our method with

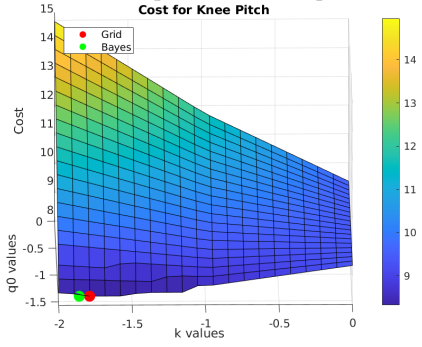
a grid search that find the k and q combinations that most reduce the torques the joints need to produce. In Figure 6 we show the cost maps and compare the optimal parameters from our Bayesian Optimization approach and the grid search fro when we increase the base mass by 3kg. We see that our method successfully avoid sub-optimal combinations in the same number of observations as of the grid search (400 data points). That is because Bayesian Optimization can rely on past observations to make informed decisions to choose the next Optimization Point [19].



(a) Hip Roll Cost Map



(b) Hip Pitch Cost Map



(c) Knee Pitch Cost Map

Fig. 6: Cost maps for Robot with extra 3kg of Base Mass. Optimal design parameters from Bayesian Optimization (Green Points) and Grid Search (Red Points) are shown.

Table 1 shows the optimal parameters obtained from our method for various added base masses. Using these optimal configurations for PEA we have new actions (torques) and can establish and new control type "S" as seen in Algorithm 1.

Algorithm 1 Compute Torques

Input: $actions$
Output: $joint\ torques$
Initialize $actions\_scaled \leftarrow actions \times$ action_scale
Initialize $control\_type \leftarrow$ control_type
if $control\_type$ is "P" **then**
  $torques \leftarrow k_p \times (actions\_scaled + q_{\text{default}} - \text{q}) - k_d \times \dot{q}$
else if $control\_type$ is "V" **then**
  $torques \leftarrow k_p \times (actions\_scaled - \dot{q}) - k_d \times (\dot{q} - \dot{q}_{t-1})/\text{t}$
else if $control\_type$ is "T" **then**
  $torques \leftarrow actions\_scaled$
else if $control\_type$ is "S" **then**
  Set $q_0 \leftarrow q_0^*$
  Set $k \leftarrow k^*$
  $spring\_torque \leftarrow (k \times (actions - q_0))$
  $torques \leftarrow actions\_scaled + spring\_torque$
  break
end if=0

C. Curriculum Learning

We know consider this new spring torque to the policy's actions under curriculum learning. In 10 000 iterations for training the policy the baseline (0 kg added mass) converged to 2.5083 m/s while our new policy converged to 2.9876 m/s. This is a increase of 19.11% in forward velocity. The number of iterations was cut at 10 000 iterations since past this point the velocity didn't significantly change.

D. Cost of Transportation

To test the performance of the approach we compared the Cost of Transportation (COT) defined as the total power over an interval of time while the positive COT refers to considering only the positive powers produced by the total sum of power from joints [22].

$$\text{COT} = \frac{\Delta W}{\text{mg}\Delta x} = \frac{1}{\text{mg}\Delta x} \sum_{j \in r,l} \int_{t_0}^{t_{\text{end}}} \left(F_{a,j}(t) \dot{h}_j(t) \right) dt \quad (11)$$

$$\text{COT}_{\text{positive}} = \frac{1}{\text{mg}\Delta x} \sum_{j \in r,l} \int_{t_0}^{t_{\text{end}}} \max \left(F_{a,j}(t) \dot{h}_j(t), 0 \right) dt \quad (12)$$

In equation (2), ΔW is the total power in a time interval, m is the robot mass, g is the acceleration due to gravity, Δx is the displacement, $F_{a,j}(t)$ represents the force of each actuator, and $\dot{h}_j(t)$ denotes the velocity of displacement for each unit of space. Equation (3) depicts the COT formula, modified to consider only positive power values.

We run the policy on 50 parallel robots and averaged the COT and positive COT at various velocities. From Fig. 6 and Figure 7. it is clear that the new policy with PBRS rewards and the found optimal PEAs lead to higher COT at lower masses until the turning point, at 1kg for regular COT and at 2kg for Positive COT. After said turning points our policy leads to lower COT, the residual torque that the motor needs to produce is smaller with our method. That is because the robot

TABLE I: Optimal PEA Parameters

Extra Mass	Hip Roll		Hip Pitch		Knee Pitch	
	k^*	q_0^*	k^*	q_0^*	k^*	q_0^*
1	-0.3729	0.1313	-1.7645	0.4433	-1.7502	-0.3176
2	-0.0253	0.0384	-0.1025	-0.3300	-1.2504	-1.5699
3	-0.0527	-1.4561	-0.0587	0.9092	-1.8602	-1.5700

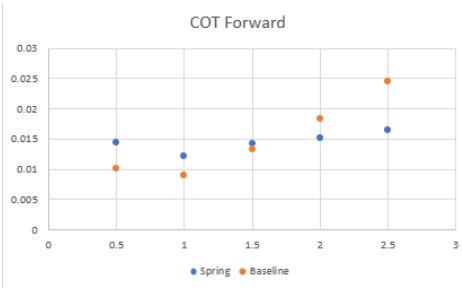


Fig. 7: Cost of Transportation

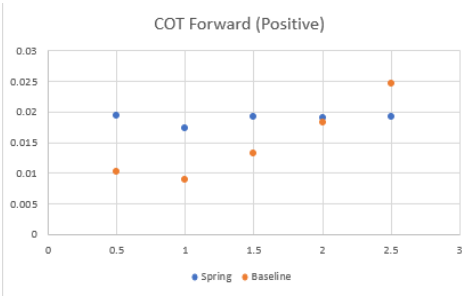


Fig. 8: Positive Cost of Transportation

can rely on the designed springs to bounce and encourage forward motion which otherwise is an additional effort at slow speeds.

V. CONCLUSION

In this work, we proposed an approach to find optimal parameters for Parallel Elastic Actuators (PEA) in Legged Robots. To generalize our approach over the various joints (Hip Pitch, Hip Roll, Knee Pitch) we first designed a Policy using Potential Based Rewards and a symmetry reward. We use the time series data from the roll-out of the policy to design a cost function relating maximum torque and energy consumption. Additionally, to evaluate performance we compared our new policy using curriculum learning for max forward velocity and compared the cost of transportation.

We also provide considerations in why Potential Based Rewards encourage velocity convergence. Using this new control applying joint stiffness Bolt Robot successfully could move move forwards faster and reduce its energy consumption at high speeds. Thus we can confirm that the effect of PEA is nontrivial since the actuators s have to repeatedly work against the spring [10]. Future works can use sample trajectories from model based methods or include vision to model the gait of Bolt when facing an obstacle and still make full use of virtual joints in various directions. Lastly, our method limited to consider joint stiffness and initial joint position in an offline fashion. It would be more ideal to consider other design parameters for different joints to see their effects as well as carrying out this optimization using Genetic Algorithm or as a Model Predictor as part of the Policy Network.

REFERENCES

[1] Fu, Z., Kumar, A., Malik, J., & Pathak, D. (2021). Minimizing Energy Consumption Leads to the Emergence of Gaits in Legged Robots. Conference on Robot Learning.

[2] Kormushev, P., Ugurlu, B., Calinon, S., Tsagarakis, N. G., Caldwell, D. G. (2011). Bipedal walking energy minimization by reinforcement learning with evolving policy parameterization. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2962-2968). doi: 10.1109/IROS.2011.6094427

[3] Sangbae Kim; Patrick M. Wensing, Design of Dynamic Legged Robots , now, 2017.

[4] A. M. Abate, "Mechanical design for robot locomotion," Ph.D. disser tation, Oregon State University, 2018.

[5] A. Mazumdar et al., "Parallel Elastic Elements Improve Energy Efficiency on the STEPPR Bipedal Walking Robot," in *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 898-908, April 2017, doi: 10.1109/TMECH.2016.2631170.

[6] K. G. Gim, J. Kim and K. Yamane, "Design of a Serial-Parallel Hybrid Leg for a Humanoid Robot," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 2018, pp. 6076-6081, doi: 10.1109/ICRA.2018.8460733.

[7] Ficht, G., Behnke, S. Bipedal Humanoid Hardware Design: a Technology Review. Curr Robot Rep 2, 201–210 (2021). https://doi.org/10.1007/s43154-021-00050-9

[8] Bauer, F., Römer, U., Fidlin, A. et al. Optimization of energy efficiency of walking bipedal robots by use of elastic couplings in the form of mechanical springs. Nonlinear Dyn 83, 1275–1301 (2016). https://doi.org/10.1007/s11071-015-2402-9

[9] T. Chen, Z. He, and M. Ciocarlie, "Hardware as policy: Mechanical and computational co-optimization using deep reinforcement learning," in Proceedings of the 2020 Conference on Robot Learning, ser. Proceedings of Machine Learning Research, vol. 155. PMLR, 16–18 Nov 2021, pp. 1158–1173.

[10] F. Bjelonic et al., "Learning-Based Design and Control for Quadrupedal Robots With Parallel-Elastic Actuators," in IEEE Robotics and Automation Letters, vol. 8, no. 3, pp. 1611-1618, March 2023, doi: 10.1109/LRA.2023.3234809.

[11] M. Hutter, et al., "Anymal-a highly mobile and dynamic quadrupedal robot," in IEEE/RSJ International conference on intelligent robots and systems (IROS), 2016, pp. 38–44

[12] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," Science robotics, vol. 5, no. 47, p. eabc5986, 2020.

[13] A. I. Cowen-Rivers, et al., "Hebo: Pushing the limits of sampleefficient hyper-parameter optimisation," J

[14] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine. Learning to walk via deep reinforcement learning. arXiv preprint arXiv:1812.11103, 2018.

[15] T. Miki et al., "Learning robust perceptive locomotion for quadrupedal robots in the wild," Science Robotics, vol. 7, no. 62, 202

[16] Rudin, Nikita & Hoeller, David & Bjelonic, Marko & Hutter, Marco. (2022). Advanced Skills by Learning Locomotion and Local Navigation End-to-End. 2497-2503. 10.1109/IROS47612.2022.9981198.

[17] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in Proceedings of the 5th Conference on Robot Learning, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 91–100. [Online]. Available: https://proceedings.mlr.press/v164/rudin22a.html

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," CoRR, vol. abs/1707.06347, 2017

[19] S. Koziel and X.-S. Yang, Computational optimization, methods and algorithms. Springer, 2011, vol. 356

[20] R. Martinez-Cantin, "Bayesian optimization with adaptive kernels for robot control," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 3350-3356, doi: 10.1109/ICRA.2017.7989380.

[21] Jeon, S. H., Heim, S., Khazoom, C., and Kim, S. (2023). "Benchmarking Potential Based Rewards for Learning Humanoid Locomotion." In 2023 IEEE International Conference on Robotics and Automation (ICRA 2023).

[22] Koseki S, Kutsuzawa K, Owaki D and Hayashibe M (2023) Multimodal bipedal locomotion generation with passive dynamics via deep reinforcement learning. Front. Neurorobot. 16:1054239. doi: 10.3389/fnbot.2022.1054239