Optimum Design of a Compliant Foot for a Quadruped

Pramod Pal
pramodpal@iisc.ac.in
Department of Mechanical Engineering,
Indian Institute of Science, Bangalore, India

Anubhab Dasgupta
anubhab.dasgupta@jgpian.iitkgp.ac.in
Department of Mechanical Engineering,
Indian Institute of Technology, Kharagpur, India

Shishir Kolathaya
shishirk@iisc.ac.in
Robert Bosch Centre for Cyber Physical Systems,
Indian Institute of Science, Bangalore, India

Ashitava Ghosal
asitava@iisc.ac.in
Department of Mechanical Engineering,
Indian Institute of Science, Bangalore, India

ABSTRACT

Improving the mechanical design of a quadruped in order to improve its performance and efficiency is an area of active research. The design of the feet with compliant element, such as a spring, can play an important role in the performance of a quadruped robot. In this work, we obtain the optimum value of the spring stiffness which results in efficient hopping behavior of a single leg using an evolutionary strategy. We start with an initial set of controller gain parameters for the hip and knee actuators along with the foot spring stiffness. The performance of each candidate solution was evaluated based on the maximum hop height. The algorithm is able to efficiently search the high-dimensional solution space and find the optimal control parameters, resulting in an improved hopping mechanism. The optimized controller gain parameters with a specific range of foot spring stiffness values showed that the compliant element at the foot helps the leg hop higher than a rigid foot.

KEYWORDS

Quadruped, Single-leg hopping, CMA-ES Evolutionary Strategy, Compliant foot

ACM Reference Format:
Pramod Pal, Anubhab Dasgupta, Shishir Kolathaya, and Ashitava Ghosal. 2023. Optimum Design of a Compliant Foot for a Quadruped. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Legged robots have gained a lot of interest and significance in recent years due to their potential applications in many different fields. Legged robots can climb, walk, and jump, making them suited for activities requiring movement across challenging or uncertain environments [4, 7]. Legged robots can mimic human movements, making them perfect for activities that require interaction with people, such as search and rescue operations or helping those who have mobility impairments [18]. Designing a quadruped robot necessitates a broad range of considerations, including the design of the legs, actuators, control systems, power supply, and overall structure. The leg design must offer stability and mobility, with the ideal number and configuration of joints, actuation type, and foot design depending on the intended use and environment.Therefore, it is essential to investigate foot design that may provide stability, terrain adaptability, increased energy efficiency, and less impact force of the robot movement on the environment, making it safer to use in populous or sensitive locations. Recent years have seen an increase in research and development of compliant feet which are intended to be flexible and deform under external stresses and this is expeted to improve the robot's performance across various terrains and situations.

Legged locomotion experiences energy losses for three main reasons: transmission losses due to friction, actuator losses due to heating, and system-environment interaction losses. Reducing the first two losses requires a new actuator design and electronics, which can be system-specific and challenging to implement [24]. More sophisticated series elastic actuators (SEAs) can solve the latter issue. By placing compliant parts in series with the actuator, SEAs reduce the ground impact force. However, SEAs are typically more expensive, complicated, and power-consuming than conventional actuators. They also have limited bandwidth and force range and can have limited accuracy due to mechanical compliance and control electronics [15]. Other forms of advanced actuation, like quasi-direct drive (QDD) actuators with a lower gear ratio, provide direct torque input and exhibit good control. However, these motors are expensive and have poor torque output with increased Joule heating [11].

Consequently, it is more practical to use a simple passive spring to store energy during the foot's contact and then release it during the other phases of the leg's motion. This not only increases the stability but also increases the efficiency of the legged robot [5]. The linear springs may be designed to meet the needs of a given application by considering parameters such as the robot's mass, size, and operating environment. It may also be engineered to produce damping to modify the rate at which the foot returns to its original position after deformation. This plays a crucial role in limiting the transfer of vibrations from the ground to the robot's frame. Linear springs are a cost-effective solution due to their versatility, durability, low material cost, and low manufacturing complexity. They are also adaptable to robots of varying sizes and weights because of their scalability. This paper will examine the effect of

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

compliant feet that use linear springs and aim to determine optimal control of the actuators along with the optimal foot stiffness.

The paper is organized as follows: in the rest of this section, we review the relevant literature. We present the leg design and its kinematics in section 2. We present the MuJoCo simulation parameters, control strategy, and optimization algorithms used for training the single-leg robot for hopping in section 3. We present the numerical simulation results in section 4, with the conclusion in section 5.

1.1 Related Work

Raibert was an early innovator in legged robotics, and his research on compliant feet for robot legs impacted the development of modern mono-ped, biped, and quadruped robots. In the studies by him and his co-workers, they focused on using passive dynamics to improve legged movement by introducing compliant features into the design of the foot. This allowed the robots to adapt to uneven terrain and recover from disturbances with less effort (see, for example, [26]). Buehler and co-workers developed hopping and quadruped running robots by incorporating springs in the robot's leg axis or hip joint [1, 21]. Kostamo et al. introduced a novel method for reducing bounce between a robotic leg and the floor using a semi-active responsive foot [16]. Hyon and Mita demonstrated that the spring converts the robot's kinetic energy to elastic potential energy and stores impulse energy for the next step [14].The 'KOLT' leg design by Palmer et al. showed minimal impact loss, low inertia, variable stiffness, and an energy storage system [20]. Focchi demonstrated the effect of variable spring stiffness by plotting the torque produced by the impact in the knee joint when the leg hits the floor [8].

2 LEG DESIGN AND CONSTRAINTS FOR HOPPING

The leg design has the following main components: the length of the links, the shape of the links, hip and knee joint placements, the actuator power ratings, and the foot design consisting of the shape of the part touching the ground and the added compliance mechanism. The ideal leg length for a jumping robot will vary depending on various elements, such as the robot's size and weight, the power of its actuators, and the environment in which it will be used-including surface friction coefficients, the incline of the surface, and so on. We have chosen to keep the shape of the links straight to keep the structure simple for analysis in the simulation. Both the actuators for the hip and knee joints were fixed to the top part of the hip link. The shank link is driven by a belt and pulley drive with a gear ratio of 1:2. For power transmission from the motor to the links, we have used a planetary gearbox with a gear ratio of 1:6. The foot is designed by adding a linear spring and damper which connects the lower extreme of the shank link to the foot part as shown in Fig. 1.

2.1 Selection of Link Lengths

In general, longer link lengths might give the robot a higher mechanical advantage since they enable a longer stride and more leverage. Larger link lengths will increase stability during the leap, but they will also require more power to get the robot off the ground and more sophisticated control algorithms to keep it there. As the literature shows, longer legs could potentially be more vulnerable to harm or failure[2]. We decided to use economical 360 KV brushless motors in our hardware and this limited the torque values. To make our system simple, we assumed equal links of 0.17 m length, which is equivalent to a medium size dog leg length [9, 27]. The base frame of the leg is allowed to move vertically in the Z-direction. The base of the leg is connected to bearing shafts with the help of linear bearing, as shown in Fig. 1, thus constraining the base motion in the Z-direction.

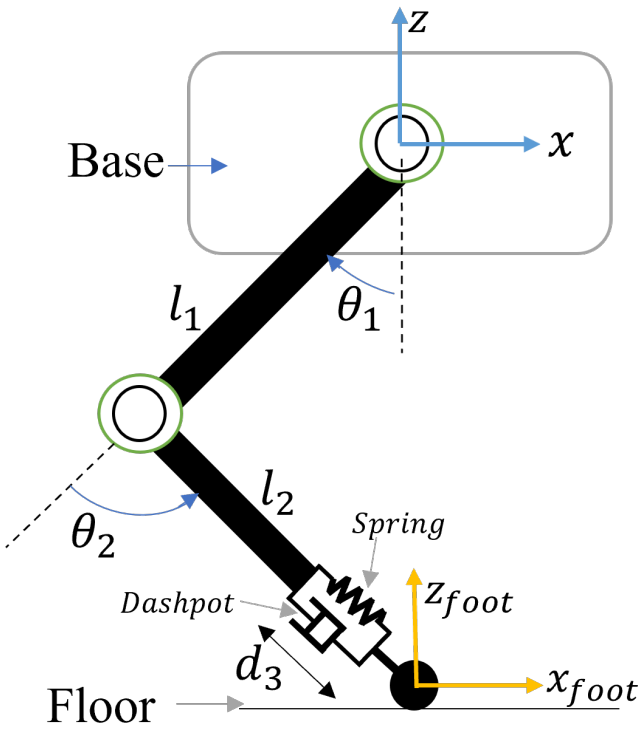

Figure 1: Schematic diagram of the single leg with compliant foot

2.2 Kinematic Modelling and Tracking Trajectory Generation

The leg is a critical component of any quadruped or a walking robot. A well-designed leg should be able to absorb sudden impact force to prevent its parts from breaking. This may be accomplished by simply attaching a compliant link to the foot, which can minimize the force of contact and simultaneously store and transfer energy while executing various gaits. The single leg, as seen in Fig. 1, consists of a base, thigh, shank, and foot link joined by two active revolute joints and one passive prismatic joint. The lengths of the thigh and shank links are represented by l_1 and l_2; the rotations

of the links are represented by the angles θ_1 (from vertical) and θ_2 (from the direction along thigh link) and their range is listed in Table 1. The forward kinematics equations provide the Cartesian locations that can be tracked by the foot tip and are given as

$$x_{foot} = -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) - d_3 \sin(\theta_1 + \theta_2) \quad (1)$$

$$z_{foot} = -l_1 \cos(\theta_1) - l_2 \cos(\theta_1 + \theta_2) - d_3 \cos(\theta_1 + \theta_2) \quad (2)$$

where d_3 denotes the linear motion of the spring (see Fig. 1).

The cyclic trajectory is made linear by constraining one of the coordinates to be a fixed point, and the resulting reference path is used to approximate the foot end-point trajectory to make the leg hop in one place. The end-point trajectory is given by

$$x = 0, \qquad z = -0.174 + 0.026 * \sin(\phi)$$

where the angle ϕ is used to divide the trajectory into 200 equally spaced points.

There are three joint variables in a leg, and to obtain θ_1, θ_2, and d_3 for a given (x, z), a redundancy resolution scheme needs to be used. In this work, we have used a sequential least squares programming (SLSQP) algorithm, originally implemented by Kraft [17] (see also [10]). We use the quantity $||\mathbf{q}||_2{}^2$ as the objective function in the optimization, where $\mathbf{q}$ denotes the vector of the hip angle, knee angle, and the change in length divided by the original length of the spring-damper system, to obtain the inverse kinematics solution of the compliant legged system.

Joint Variable	Joint name	Type of joint	Range
θ_1	hip	Revolute	0 to 1.6 radian
θ_2	knee	Revolute	-2.1 to 0 radian
d_3	foot	Prismatic	0 to 0.012 m

Table 1: Joint variables motion range

3 MUJOCO SIMULATION AND PARAMETERS USED

To study the effect of linear spring at foot, we made use of the open AI MuJoCo simulation environment [25]. MuJoCo captures contact dynamics with state-of-the-art capabilities. It is a popular open-source program for robotic system simulation and is simple to use. To communicate with the simulator and get experimental data, we used the MuJoCo-py module. In reference [6], MuJoCo is shown to be the quickest and most accurate for robotics-related system. The model shown in the Fig. 2 represents the single leg that has been designed using the Solidworks program, and then the URDF (Unified Robotics Description Format) is created with the help of an add-on module. Taking reference from the URDF file for the position and orientation of the links, an XML file world body tag is made. Using XML API documentation, we first make the ground plane on which the single leg will jump and then added other leg elements along with the base frame.

To correctly depict the robot's behavior using the MuJoCo simulation, a number of parameters must be specified. These parameters consist of model parameters, which determine the physical qualities of the simulation's objects, such as mass, size, and location. Simulation parameters, such as the time step size and the number of iterations, govern the simulation itself. Control parameters, such as actuator forces and torques, define how the robot is controlled. The visualization factors, such as camera position and lighting conditions, dictate how the simulation can be shown. Solver parameters can be used to control the numerical solver, which is utilized to solve the equations of motion and the stability and precision of the simulation can be controlled with the use of these parameters. Environment parameters describe the attributes of the simulated environment, such as the coefficients of gravity and friction. All of these parameters work together to form a realistic depiction of the robot and its environment, enabling the realistic simulation and assessment of robotic systems.

In our simulations, we assigned a gravity that allows the single leg to fall to make contact with the ground when it hits the ground plane. In MuJoCo, a variety of actuators are available. We have used the proportional and derivative controller gain of a motor to simulate the behavior of a spring and damper as an added compliance mechanism to the foot of the single-legged robot. We have used a timestep of 0.001 which provides better accuracy and stability. For the solver, we have used Newton's method. A 4^{th}-order Runge-Kutta method is used as an integrator, known to be better than the Euler method for the energy-conserving system both in terms of stability and accuracy [23]. The total mass of the single leg is 3 kg. At all joints, we have chosen a damping of 1 N s/m. The torque is limited in both hip and knee joints to 7 N-m.

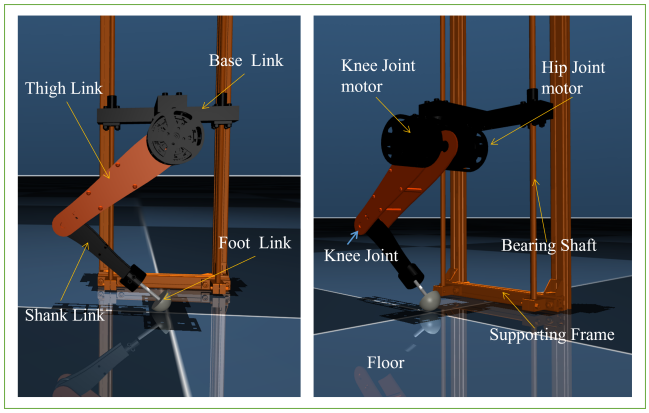

Figure 2: Single leg with compliant foot in MuJoCo environment

Fig. 3 shows the block diagram of simulation. We first find the optimal proportional (k_p) and derivative gains (k_d) for hip and knee joints motor with the help of a data-driven evolutionary algorithm – see section 3.2 below for the details of the three different evolutionary strategies that were attempted. These gains are applied to the errors of the joint state to yield torque commands. The torques are given to the drive motors to yield new joint states and the base position. The base position helps to compute the cost function, and the evolutionary strategy algorithm maximizes the cost function by maximizing the jump. We trained the model for a range of spring

stiffness to get optimal k_p and k_d for the hip and knee joint motor. We then chose the spring stiffness which gives the maximum cost function value.

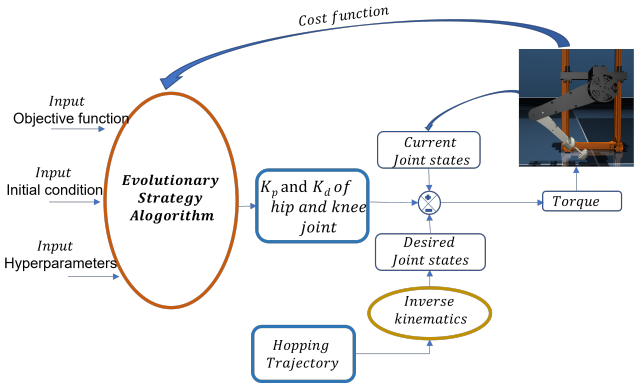

Figure 3: Schematic diagram of single leg with compliant foot

3.1 Control Strategy

In the simulation, torque is used as the control input. The control system uses the proportional plus derivative (PD) strategy to calculate the necessary torque output based on the current position and velocity of the system as well as the desired position and velocity – see Fig. 2. The PD control continues to be one of the most popular strategies despite significant advancements in nonlinear control systems and development of model-based controllers. PD controllers draw more community interest since they don't require models (and its inherent uncertainity) and are simpler to develop. In walking robots like bipeds, PD controllers have been widely employed, and research has shown that these controllers are locally stable in a variety of circumstances. The PD controller for the three actuators can be written as

$$\tau_1 = k_{p_1} e + k_{d1} \dot{e}, \quad \tau_2 = k_{p_2} e + k_{d2} \dot{e}, \quad \tau_3 = k_{p_3} e + k_{d3} \dot{e}$$

where, τ_1,τ_2,τ_3, are the torques k_{p_1}, k_{p_2}, k_{p_3} are the proportional gains, k_{d1}, k_{d2}, k_{d3} are the derivative gains, e is error in current position and desired position and $\dot{e}$ is rate of change of error, respectively.

3.2 Evolutionary Algorithms

We have implemented three distinct evolutionary strategies to find the optimal controller gains for the single-legged system. A brief description of each of them is given below:

- **Covariance Matrix Adaptation Evolutionary Strategy**: We have implemented the work of Hansen et al. [13] . In the CMA-ES evolution strategy, new candidate solutions are sampled according to a multivariate normal distribution in $\mathbb{R}^n$x$\mathbb{R}^n$. In our case $n = 6$ comprising of the hip, knee, and foot k_p, k_d values. We treat the problem as a black box optimization problem, and thus the optimization is model-free. We sample controller gains from the multivariate normal distribution and perform a mutation operation. Mutation amounts to adding a random vector, a perturbation with zero mean. Pairwise dependencies between the variables in the distribution are represented by a covariance matrix. The covariance matrix adaptation (CMA) is a method to update the covariance matrix of this distribution, which gives the next distribution with an updated covariance matrix and mean, from which the following iteration's controller gains will be sampled from. CMA-ES uses the following as its equation for sampling:
$$\mathbf{k} \leftarrow m + \sigma \mathbf{k}^{'} \sim \mathcal{N}(m, \sigma^2 C)$$
where $\mathbf{k}^{'} \sim \mathcal{N}(0, C)$. Following the above, the covariance matrix C is updated, and the mean vector m is updated using linear weighted recombination followed by step-size control.
- **Elitist Genetic Algorithm**: For comparison, we implemented the Elitist Genetic Algorithm, where the method keeps the best individuals of a generation alive until some better individual arrives. This ensures a monotonic evolution of the cost function. It uses mutation and cross-over operations to optimize the objective, which is the cost function of the environment setup. Bhandari showed EGA converges to the global optimal solution with any choice of initial population [3], Raji showed that it only takes a small number of fitness values to converge [22]. In this work, we observed that the CMA-ES algorithm performed better than this algorithm in most simulations.
- **Cartesian Genetic Programming**: We have also implemented Cartesian Genetic Programming [19], which shows the usage of a particular tree-based functional expression encoding to represent the parameters to be optimized – in this case the controller gains. Cross-over and mutation operations are then performed to update the encoding parameters, which finally yields the updated set of controller gains. When compared, CMA-ES outperformed Cartesian Genetic Programming, and therefore, we selected the CMA-ES algorithm for the training.

The CMA-ES is particularly useful if the reward function is ill-conditioned or non-convex over the sample space and hence the choice of using it over several other evolutionary algorithms. It is known in the literature that the CMA-ES is a prominent algorithm best suited for non-convex problems with an uneven sample space [12], the reason being that CMA-ES uses a normal distribution from which gains are sampled and they have the largest entropy for a given sample space.

3.3 Training

We have used an evolutionary training method that finds the optimal gains of the controller by using a model-free objective function. The objective function has been chosen to depend solely on the jump height from the ground of the base slider. The motivation for choosing this as an objective, and not something like the range of jump, is because local maximization of the range is possible

even if the base slides down to the ground and moves up minimally. The selection of this objective function is also motivated by the fact that it is a convex function in h and that the speed of CMA-ES search is not greatly slowed by its use. In this work, we chose $Ke^{-(\mathbb{H}-h)}$ as an objective to be maximized, where K is a non-zero positive constant, $\mathbb{H}$ represents an upper limit on the jumping height (assumed to be much bigger than what the robot can jump realistically), and h is the height of the base-slider in the step in which it is getting computed. We took 4000 steps and calculated the objective as $max(R_i) \in [2000 \rightarrow 4000]$, where R_i is the objective function value at the i-th iteration of the training. This preserves the nature of the step-wise objective without affecting the end goal of the training.

4 RESULTS AND DISCUSSION

In this section, we present some of the main simulation results. We also present the training results and the usage of the trained controller gains to analyze the hopping behavior.

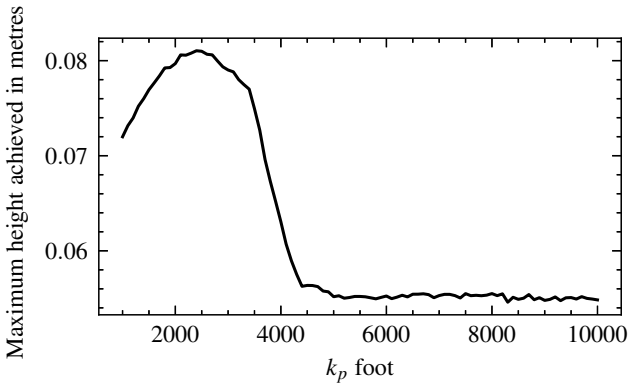

Figure 4: Maximum base height achieved with varying foot spring constant for fixed values of controller gains obtained after training

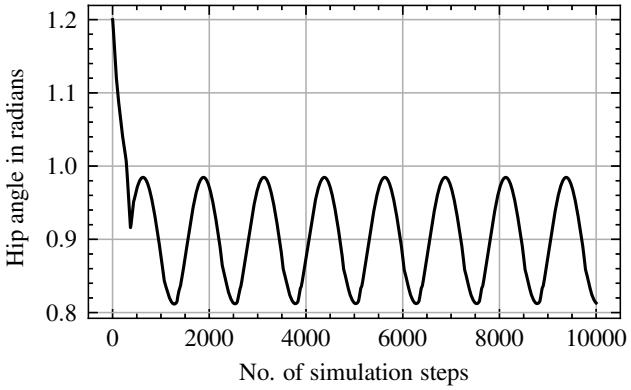

Figure 5: Hip angle with number of simulation time steps

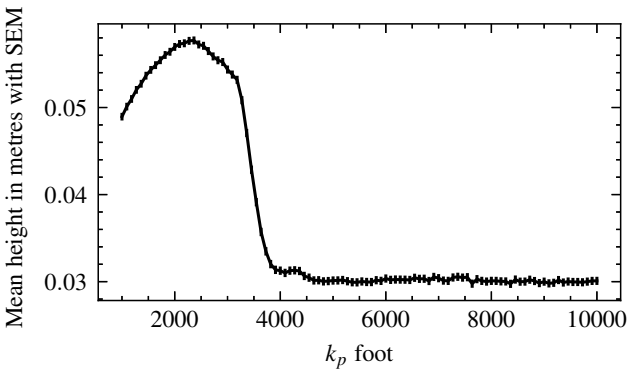

Figure 6: Mean height of base slider with standard error of the mean ($\pm 10^{-4}$)

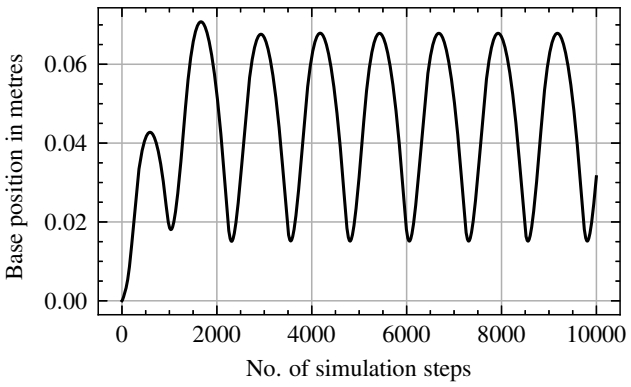

Figure 7: Height of base slider at 1000 k_p foot

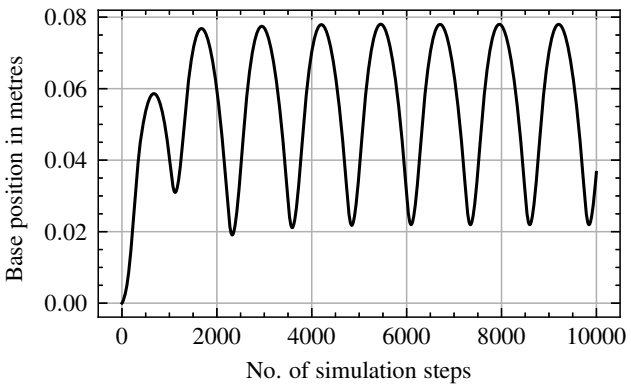

Figure 8: Height of base slider at 2400 k_p foot

The single-leg hopping is simulated for a range of spring stiffness (1000 N/m to 10000 N/m) in MuJoCo shown in Fig. 4. This graph is generated by varying the stiffness of the foot joint, keeping values of controller gains and other parameters fixed obtained after training. This graph clearly shows that the maximum height achieved by

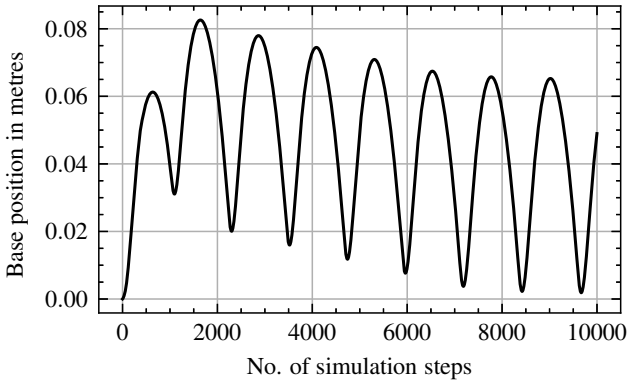

Figure 9: Height of base slider at 8000 k_p foot

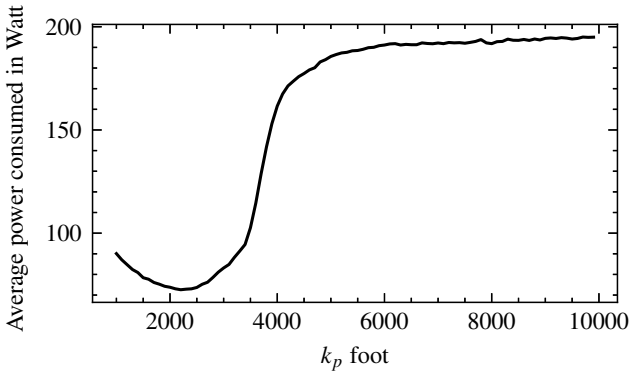

Figure 10: Average power consumed with foot stiffness

the base link keeps on increasing until an optimum stiffness of the foot link. After the optimal stiffness, the maximum base link height decreases and gets saturated at very high stiffness values (fixed foot). Fig. 5 shows the variation of the hip angle with the number of simulation time steps, where the initial transient behavior of jumping down from a higher hip angle to a periodic lower value is due to the system trying to position itself at the correct hopping coordinates. As we have seen in multiple simulation results, the periodic behavior is independent of the initial starting position of the leg, which demonstrates the stability and robustness of the controller with respect to initial joint states.

Fig. 8 shows the variation of the base position with simulation time steps and this demonstrates that the base varies periodically with almost the same amplitude. It gradually increases and stabilizes itself about a particular mean. Fig. 6 shows the mean height of the base with varying spring constant of the foot with the error scales representing the deviation of the cumulative mean. The increasing error scales at the end demonstrate the fact that the standard deviation increases with the foot k_p up to a particular value and then decreases again. As is evident from Fig. 6, the standard deviation achieves a peak value at a k_p of the foot at around 2400 which is the region of maximum hopping range for the particular value of the controller gains. Fig. 7 shows the base position keeping the foot k_p at 1000. The maximum height achieved by the base link is 0.071 m which is low compared to the optimal k_p of 2400 shown in Fig 8 where the height reached is 0.081 m. We also obtained the height achieved for a high k_p of 8000, and as shown in Fig. 9 a maximum value of 0.054 m was achieved. Fig. 10 shows the average power consumed as a function of foot stiffness and it can be see that the power consumed is least for the optimum foot stiffness.

The overall the simulation results indicate that the trained controller is robust with respect to the initial position, and the base height from the ground gradually increases over time and stabilizes itself at a particular mean. There exists a local supremum in the maximum height achieved by the base slider as shown in Fig. 4 for the given inertial parameters and the trained controller gains, which demonstrates the existence of a particularly favorable range of foot spring-damper values for which the single leg demonstrates maximum hopping ability.

5 CONCLUSION

In this work, the effect of foot compliance on a single-legged hopping robot has been studied. The single leg has two motors and a spring at the feet, and the objective was to use evolutionary techniques to find the best combination of parameters for the highest vertical hop. MuJoCo a rigid body simulator, is used to study the effect of linear spring at the foot on the single-legged robot model. The primary finding of the study is that there is a stiffness and control gain range where the vertical jump is maximum. According to the simulation data, the optimum spring stiffness at the foot can enhance the highest point that can be reached by the base. This work is being extended to optimize the impact force and power consumption and to implement the findings on a physical hardware. The main golal is to apply this research to a quadruped, where stiffness will be added to each of the four legs to examine how the stiffness of the feet affects the quadruped's cost of transport.

REFERENCES

[1] Mojtaba Ahmadi and Martin Buehler. 2006. Controlled passive dynamic running experiments with the ARL-monopod II. *IEEE Transactions on Robotics* 22, 5 (2006), 974–986.
[2] Rafhael M Andrade and Paolo Bonato. 2021. The role played by mass, friction, and inertia on the driving torques of lower-limb gait training exoskeletons. *IEEE Transactions on Medical Robotics and Bionics* 3, 1 (2021), 125–136.
[3] Dinabandhu Bhandari, CA Murthy, and Sankar K Pal. 1996. Genetic algorithm with elitist model and its convergence. *International Journal of Pattern Recognition and Artificial Intelligence* 10, 06 (1996), 731–747.
[4] Priyaranjan Biswal and Prases K Mohanty. 2021. Development of quadruped walking robots: A review. *Ain Shams Engineering Journal* 12, 2 (2021), 2017–2031.
[5] Jie Chen, Zhongchao Liang, Yanhe Zhu, Chong Liu, Lei Zhang, Lina Hao, and Jie Zhao. 2019. Towards the exploitation of physical compliance in segmented and electrically actuated robotic legs: A review focused on elastic mechanisms. *Sensors* 19, 24 (2019), 5351.
[6] Tom Erez, Yuval Tassa, and Emanuel Todorov. 2015. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 4397–4404.
[7] Jorge Ferreira, A Paulo Moreira, Manuel Silva, and Filipe Santos. 2022. A survey on localization, mapping, and trajectory planning for quadruped robots in vineyards. In *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 237–242.
[8] Michele Focchi. 2013. Strategies to improve the impedance control performance of a quadruped robot. *Genoa: Istituto Italiano di Tecnologia* (2013).
[9] Elena Garcia, Juan Carlos Arevalo, Gustavo Munoz, and Pablo Gonzalez-de Santos. 2011. On the biomimetic design of agile-robot legs. *Sensors* 11, 12 (2011), 11305–11334.
[10] Ashitava Ghosal. 2006. *Robotics.* Oxford University Press.

[11] Gianluigi Grandesso, Gabriel Bravo-Palacios, Patrick Wensing, Marco Fontana, and Andrea Del Prete. 2020. Exploring the limits of a hybrid actuation system through Co-Design. (2020).

[12] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. 2010. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*. 1689–1696.

[13] N. Hansen, S.D. Muller, and P. Koumoutsakos. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation* 11, 1 (2003), 1–18.

[14] Sang-Ho Hyon and Tsutomu Mita. 2002. Development of a biologically inspired hopping robot-" Kenken". In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, Vol. 4. IEEE, 3984–3991.

[15] AG Leal Junior, Rafhael Milanezi de Andrade, and Antônio Bento Filho. 2016. Series elastic actuator: Design, analysis and comparison. *Recent Advances in Robotic Systems* (2016).

[16] Esa Kostamo, Michele Focchi, Emanuele Guglielmino, Jari Kostamo, Claudio Semini, Jonas Buchli, Matti Pietola, and Darwin Caldwell. 2014. Magnetorheologically damped compliant foot for legged robotic application. *Journal of Mechanical Design* 136, 2 (2014), 021003.

[17] Dieter Kraft. July,1988. *A software package for sequential quadratic programming*. Technical Report DFVLR-FB 88-28, Institut für Dynamik der Flugsysteme, Oberpfaffenhofen.

[18] Yibin Li, Bin Li, Jiuhong Ruan, and Xuewen Rong. 2011. Research of mammal bionic quadruped robots: A review. In *2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE, 166–171.

[19] Julian Miller. 2003. *Cartesian Genetic Programming*. Vol. 43. https://doi.org/10.1007/978-3-642-17310-3

[20] Luther R Palmer, David E Orin, Duane W Marhefka, James P Schmiedeler, and Kenneth J Waldron. 2003. Intelligent control of an experimental articulated leg for a galloping machine. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, Vol. 3. IEEE, 3821–3827.

[21] Ioannis Poulakakis, James Andrew Smith, and Martin Buehler. 2005. Modeling and experiments of untethered quadrupedal running with a bounding gait: The Scout II robot. *The International Journal of Robotics Research* 24, 4 (2005), 239–256.

[22] Ismail Damilola Raji, Habeeb Bello-Salau, Ime Jarlath Umoh, Adeiza James Onumanyi, Mutiu Adesina Adegboye, and Ahmed Tijani Salawudeen. 2022. Simple Deterministic Selection-Based Genetic Algorithm for Hyperparameter Tuning of Machine Learning Models. *Applied Sciences* 12, 3 (2022), 1186.

[23] Nikita Rudin, Hendrik Kolvenbach, Vassilios Tsounis, and Marco Hutter. 2021. Cat-like jumping and landing of legged robots in low gravity using deep reinforcement learning. *IEEE Transactions on Robotics* 38, 1 (2021), 317–328.

[24] Sangok Seok, Albert Wang, Meng Yee Chuah, Dong Jin Hyun, Jongwoo Lee, David M Otten, Jeffrey H Lang, and Sangbae Kim. 2014. Design principles for energy-efficient legged locomotion and implementation on the MIT cheetah robot. *IEEE/ASME Transactions on Mechatronics* 20, 3 (2014), 1117–1129.

[25] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5026–5033.

[26] David Wooden, Matthew Malchano, Kevin Blankespoor, Andrew Howardy, Alfred A Rizzi, and Marc Raibert. 2010. Autonomous navigation for BigDog. In *2010 IEEE International Conference on Robotics and Automation*. IEEE, 4736–4741.

[27] Yuhai Zhong, Runxiao Wang, Huashan Feng, and Yasheng Chen. 2019. Analysis and research of quadruped robot's legs: A comprehensive review. *International Journal of Advanced Robotic Systems* 16, 3 (2019), 1729881419844148.