2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

Terrain-Blind Humanoid Walking on Rough Terrain with Trajectory Optimization and Biarticular Springs

1st Mustafa Melih Pelit
Systems and Control Engineering
Tokyo Institute of Technology
Tokyo, Japan
pelit@ac.sc.e.titech.ac.jp

2nd Masaki Yamakita
Systems and Control Engineering
Tokyo Institute of Technology
Tokyo, Japan
yamakita@ac.sc.e.titech.ac.jp

***Abstract*—Trajectory optimization techniques to control biped walkers are becoming popular with improvements in available solvers. However, many of the proposed controllers assume that the terrain is flat, causing the biped robot to easily fall when the assumption doesn't hold. Humans can easily walk on rough terrain and there are a number of controllers that deal with this issue through perception or sensing but necessary research to tackle this issue without perception/sensing is still lacking. If the walking controller can deal with terrain changes without perception/sensing (terrain-blind), this would ease the computational burden on the controller and decrease the problems caused by errors in perception. This paper proposes a controller that can track the optimized trajectories while handling moderate changes in terrain height. This was mainly achieved by our phase variable manipulation and utilization of a second optimized trajectory that lands the robot safely. We have also improved the robustness of the robot mechanically, by adding passive biarticular muscles. Furthermore, we investigated the effect of biarticular muscle parameters on robustness. Through simulation studies, we show that our proposed controller with proper biarticular muscle parameters can have a 5-link underactuated robot walk without falling on terrains with up to 6.47 cm height changes.**

***Index Terms*—legged robots, robot dynamics and control, bipedal robots**

I. Introduction

BIPEDAL movement is a great option for functional robots since they could ideally operate in environments that humans work in. For this reason, research in this area has gathered a lot of interest. Human gait is efficient and very robust but bipedal robots are yet to mimic this success. One reason why bipedal robots have a hard time operating in real-world environments is because of rough terrains such as gravel roads, farming fields, forests etc. This is why in this paper, we tried to improve the robustness of a 5-link underactuated bipedal robot when it moves in unknown (blind walking) rough terrain via the addition of passive biarticular muscles along with our proposed controller based on optimized trajectories.

Trajectory optimization techniques are commonly used for bipedal robots because they can provide optimal or locally optimal gaits to the chosen cost functions. They are increasingly gaining interest due to the improvements in computational power and commercially available solvers that can handle constrained non-linear problems. In [1], non-linear programming with basic splines was utilized to solve the trajectory optimization problem of a biped robot with series elastic actuators by assuming a fixed-base model. In [2], a library of optimal trajectories were used to achieve speed tracking in the 3D actuated robot ATRIAS. Optimization code used to generate the trajectories that make up the gait library is explained in [3] where they used direct transcription methods to utilize nonlinear programming solvers. In [4], direct collocation methods were used to find an optimal trajectory for a template model called spring-loaded inverted pendulum model with swing legs. Then the reference trajectories from the resulting template model was used to control a 5-link fully actuated robot. Studies mentioned above show the effectiveness of trajectory optimization methods in achieving bipedal gait but they all assume the walking occurs on a flat surface and the robustness of these methods are questionable.

One important factor of robustness for biped robots is their ability to walk over uneven or rough terrain. There are many proposed controllers that tackle the walking on known

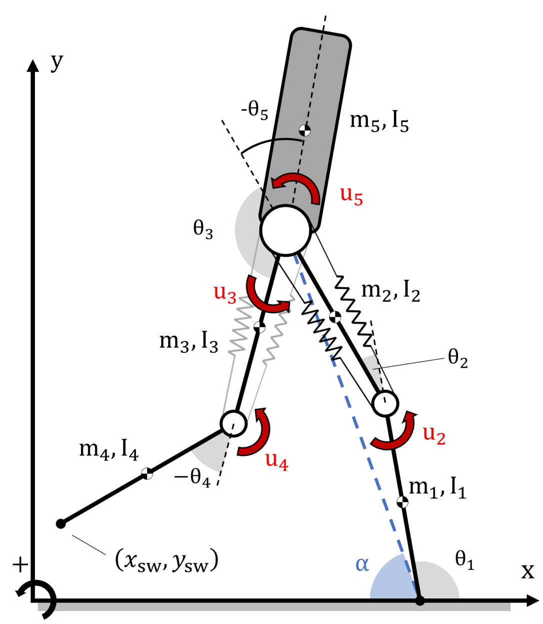

Fig. 1. 5-link underactuated bipedal robot model where the inputs are indicated with red arrows

uneven terrain issue but the ability to overcome changes in ground height without perceiving the environment is still an open research problem. If a robot is able to walk on rough terrain blindly, it would ease the burden on the controller of handling this issue and reduce the problems caused by inaccurate perception. In [5], a one step adaptation strategy based on an actuated dual-SLIP template model was proposed, so that humanoid robot can walk on rough terrain without perceiving it for terrain variations of ± 5 cm.

One way to tackle the robustness issue is to do it on a mechanical level, namely by adding compliant elements. For example in [6], they found that the series elastic compliance has a stronger impact compared to the parallel one. Also, it is indicated that a stiffer leg increases efficiency but reduces the robustness. However, compliant elements are usually implemented in monoarticular fashion or as series elastic actuation. Monoarticular muscles are connected to two links and can drive a single joint. On the other hand, biarticular muscles are connected to two links separated by a third one and they can drive two joints at the same time [7]. There are studies that indicate the benefits of these under-studied biarticular muscles.

In [8], the important contributions of biarticular muscles to trajectory control, stiffness control, and output control that take place at the extremities were illustrated. In [9], it is shown that the biarticular muscle mechanism they use in the robotic leg contributes to improved force capacity in such a way that the total output performance is maintained while individual actuator requirements are reduced. It was shown in [10] that biarticular muscles contribute to muscle coordination when performing a jumping motion.

We showed in a previous study that walking efficiency, walking speed and minimum input torque requirements can be significantly improved by using passive biarticular muscles [11]. This study was conducted using trajectory optimization methods and it illustrated that different stiffness values were needed for different optimization goals.

This paper focuses on terrain-blind walking based on optimized trajectories of a 5-link underactuated (point-footed) biped robot equipped with biarticular muscles. The primary contributions of this paper includes:

- Proposing a controller that can handle walking on a terrain with random height variations using reference trajectories without perception/sensing and validating it on simulation experiments. We achieved this by using different reference trajectories for walking and stepping-down.
- Investigating the effects of passive biarticular muscles on a bipedal robot's ability to walk in rough terrain and how different physical parameters of these biarticular springs can effect the robustness.

This paper is organized as follows. Section II describes the robot model. Section III describes the optimization setup that was used to obtain the reference trajectories. Section IV describes the proposed controller and in Section V we present our results.

II. Systems and Modeling

In this section, we will describe the 5-link underactuated bipedal robot model used in this paper and our method of generating random rough terrains.

A. Bipedal Walker with Biarticular Spring

The robot model and the notations that are used to describe it can be seen in Fig. 1. This planar model consists of 5 links representing the lower leg, the upper leg and the torso. Model is underactuated and point-footed (no ankle torque). It has 2 actuators on the knees and 2 on the hip where the revolute joints are positioned. Also, there are springs connected in a biarticular configuration between the torso and the lower legs.

Equation of motion of this model can be written as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}\mathbf{u} + \boldsymbol{\tau}, \tag{1}$$

where $\mathbf{q} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]^{\mathrm{T}} \in \mathbb{R}^5$ are the generalized coordinates, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{5\times 5}$ is the inertia matrix, $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^5$ is the Coriolis, centrifugal and gravitational terms vector, $\mathbf{S} \in \mathbb{R}^{5\times 4}$ is the distribution matrix of the inputs, $\mathbf{u} = [u_2, u_3, u_4, u_5]^{\mathrm{T}} \in \mathbb{R}^4$ are the input torques and $\boldsymbol{\tau} \in \mathbb{R}^5$ represents the torques generated by the biarticular springs. $\boldsymbol{\tau}$ can be expanded as:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathrm{st}} + \boldsymbol{\tau}_{\mathrm{sw}}, \tag{2}$$

where subscripts "st" and "sw" respectively represent the "stance leg" which is the leg that is in contact with the ground and the other leg called the "swing leg".

We calculate the biarticular spring torques in the same manner as [11] where the partial derivative of the potential energy stored in the springs is taken with respect to the generalized coordinates, resulting in:

$$\boldsymbol{\tau}_{\mathrm{st}} = \begin{bmatrix} 0 \\ -\kappa\, r_{\mathrm{k}} \Delta l_{\mathrm{st}} \\ 0 \\ 0 \\ -\kappa\, r_{\mathrm{h}} \Delta l_{\mathrm{st}} \end{bmatrix}, \ \boldsymbol{\tau}_{\mathrm{sw}} = \begin{bmatrix} 0 \\ 0 \\ \kappa\, r_{\mathrm{h}} \Delta l_{\mathrm{sw}} \\ \kappa\, r_{\mathrm{k}} \Delta l_{\mathrm{sw}} \\ -\kappa\, r_{\mathrm{h}} \Delta l_{\mathrm{sw}} \end{bmatrix}, \tag{3}$$

where κ [N/m] is the biarticular spring stiffness, r_{h} [m] and r_{k} [m] are the lever arm lengths with subscripts "h" and "k" referring to "hip" and "knee".

$$\Delta l_n = r_h(\varphi_h^n - \varphi_{h0}) - r_k(\varphi_k^n - \varphi_{k0}),\ n \in \{\mathrm{sw}, \mathrm{st}\}, \tag{4}$$

are the deflection of the respective spring. Terms related to biarticular springs are indicated in Fig. 2.

We introduce the lever arm ratio $r = r_h/r_k$, and new spring constant term $\bar{\kappa} = \kappa\, r_{\mathrm{k}}^2$ and deflection

$$\Delta \bar{l}_n = \frac{\Delta l_n}{r_{\mathrm{k}}}, \tag{5}$$

so that in Section III, we can search for r and $\bar{\kappa}$ [Nm] instead of the r_{h}, r_{k} and κ [N/m]. Equation (3) can be rewritten using these new terms.

Equation (1) models the single stance phase (when one foot is on the ground and the other is doing the swinging motion). When the swing foot contacts the ground (touch-down), an impact occurs and model goes to a double stance phase where both foot are on the ground. In this paper, we assume an instantaneous double stance phase i.e. swing leg and stance leg switch instantaneously at the moment of the impact and after the impact, swing leg lifts up from the ground without interaction (lift-off). The reset map is given by:

$$\boldsymbol{x}^{+} = f_H(\boldsymbol{x}^{-}). \tag{6}$$

where $\boldsymbol{x} = [\mathbf{q}^{\mathrm{T}}, \dot{\mathbf{q}}^{\mathrm{T}}]^{\mathrm{T}}$. During this event, position of the robot remains the same, only the swing leg and stance leg are swapped but velocities change discontinuously [11].

In this paper, we also compare the performance of our model with the default bipedal model (model without the biarticular springs). Default model can be achieved by setting the spring stiffness values to zero in Equation 1. Throughout this paper, the surface is considered rigid with sufficient friction to allow the movement.

B. Generating the rough terrain

We randomly generated rough terrains to test the performance of the proposed controller. First, numbers between 0 and 1 were randomly generated for every 0.1 [m] interval of the track length to generate a seed. Then the seed is multiplied with $\delta \in [0 : 0.001 : 0.1\,[m]]$ to set the maximum height of the terrain where $\delta = 0$ [m] is the flat terrain. Some sample terrains generated this way can be seen in Fig 3. We linearly interpolate for intermediary points in the terrain.

By increasing δ we can monotonically increase the difficulty of a certain terrain seed. This means that the controller will be able to handle the terrain with increasing δ. How the terrain looks for different values of δ can be seen in Fig. 4.

III. Optimization

In this section, we will describe the optimization setup that was used in obtaining the reference trajectories for regular walking and stepping-down motions using direct collocation method [12]. These methods turn the continuous time problem into discrete one which then can be handled by nonlinear programming solvers. In this paper, OpenOCL [13] was used to solve the trajectory optimization problem.

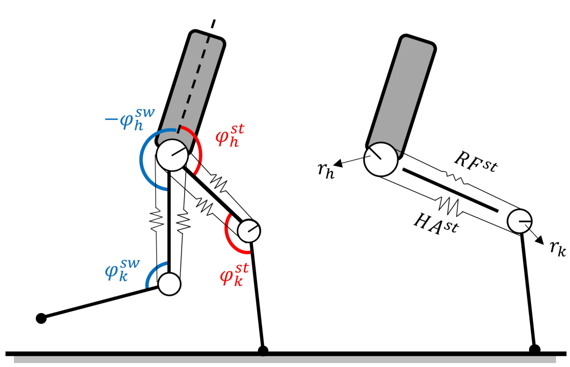

Fig. 2. Biarticular spring related variables

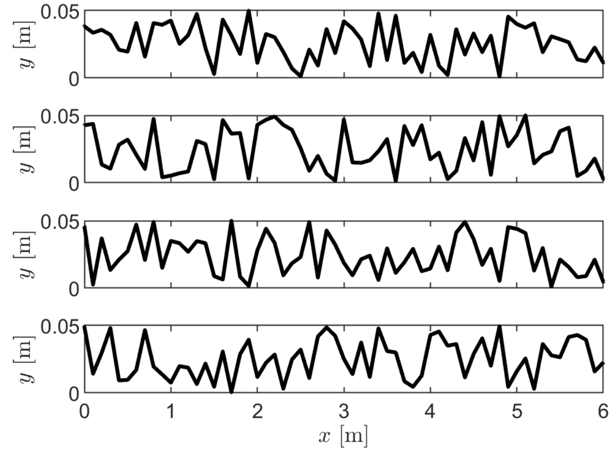

Fig. 3. Sample of randomly generated rough terrains ($\delta = 0.05$ [m])

Optimization problem can be formulated as

$$\begin{aligned} \min_{\boldsymbol{x},\mathbf{u},\boldsymbol{p},T} \quad & \int_0^T L(\boldsymbol{x}(t), \mathbf{u}(t), \boldsymbol{p})dt \\ \text{s.t.} \quad & \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \mathbf{u}(t), \boldsymbol{p}) \\ & \boldsymbol{r}(\boldsymbol{x}, t, \boldsymbol{p}) \leq 0, \end{aligned} \tag{7}$$

where $t \in [0, T]$ is the time, $\boldsymbol{x}(t)$ is the state trajectory as defined in Section II-A, $\mathbf{u}(t)$ are the inputs, $\boldsymbol{p}$ are the parameters, $L(\boldsymbol{x}(t), \mathbf{u}(t), \boldsymbol{p})$ is the path cost function, $\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{p})$ is the system dynamics function (differential equation) and $\boldsymbol{r}(\boldsymbol{x}, t, \boldsymbol{p})$ are the constraint functions. The dynamic constraints and the inequality constraints are realized on grid points (collocation points). Number of the grid points was chosen as $N = 24$ and degree of interpolating polynomial as $d = 3$ for the optimization.

We want to obtain two different trajectories for walking on rough terrain. First one is called the "walking trajectory" and this is the reference for our controller most of the time. This would be the only trajectory needed if the gait was to be performed on an even terrain. However, while walking on uneven terrain and when the robot reaches the end of the pro-

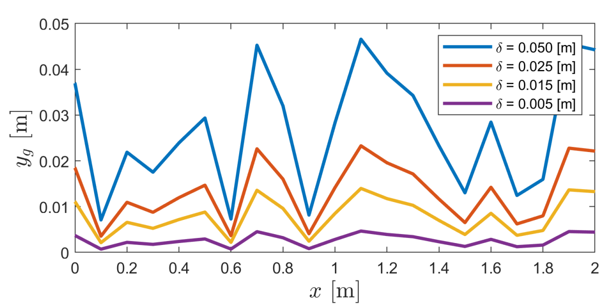

Fig. 4. Effect of increasing δ on the terrain

vided reference walking trajectory without managing to touch the ground due to terrain roughness, we need an additional reference trajectory to safely land the robot. To address this, we will also generate a "stepping-down" trajectory. If the next touch-down position is higher than the ground level, "walking trajectory" can handle it to a certain degree.

The cost function for the walking and stepping-down trajectories are identical and set to

$$L(\boldsymbol{x}(t), \mathbf{u}(t), \boldsymbol{p}) = \mathbf{u}(t)^{\mathrm{T}}\mathbf{u}(t). \tag{8}$$

For the two trajectories, we need to determine a set of constrains so that resulting trajectory is a human-like walking gait. Some constraints for the two trajectories are different but the common ones are:

- Constraining the relative knee joint angles to achieve human-like gaits: $5° < \theta_2 < 22.5°$, $270° < \theta_4 < 345°$
- Upper body must remain straight: $80° < \theta_5 < 90°$
- The angular velocity of motors must not be over the desired limit: $|\dot{\theta}_i| < 10 \ \ [\mathrm{rad}/s], i \in \{1, 2, 3, 4, 5\}$
- Center of mass of the robot should always be moving with positive velocity in the x direction: $\dot{x}_{\mathrm{CoM}}(t) > 0$ [m/s]
- Setting a lower bound for virtual stance leg angle α's velocity to keep it monotonically increasing (we set the lower bound to 0.3 [rad/s] rather than to 0 [rad/s] in order to have some safety margin in the case of blind walking in rough terrain): $\dot{\alpha} > 0.3$ [rad/s]
- The step length of the robot is set to be 0.25 [m]

Snap-shots from resulting walking and stepping-down trajectories are shown in Fig. 5.

A. Walking Trajectory

The constraints for the walking trajectory are set as follows:

- Swing foot related constraints: $y_{\mathrm{sw}}(0) = y_{\mathrm{sw}}(T) = 0$ $[m]$, $y_{\mathrm{sw}}(0 < t < T) > 0$ [m], $\dot{y}_{\mathrm{sw}}(T) < -0.2$ [m/s], $\dot{x}_{\mathrm{sw}}(T) < 0$ [m/s], $\dot{x}_{\mathrm{sw}}(t < T) > 0$ [m/s]
- The trajectory must be periodic: $\boldsymbol{x}(0) = f_H(\boldsymbol{x}(T))$ (f_H is the reset map in Equation 6)
- Mechanical parameters of the biarticular springs were constrained as: $0.01 \leq r \leq 5$ and $0 \ [Nm] \leq \bar{\kappa} \leq 2000 \ [Nm]$.
- Swing foot must avoid a virtual elliptic obstacle (prevents foot dragging and keeps to swing leg from contacting the ground early on rough terrains):

$$\left(\frac{x_{\mathrm{sw}}(t) - d_{\mathrm{obs}}}{w_{\mathrm{obs}}}\right)^2 + \left(\frac{y_{\mathrm{sw}}(t)}{h_{\mathrm{obs}}}\right)^2 \geq 1, \tag{9}$$

where x_{sw} and y_{sw} are horizontal and vertical positions of the swing foot, $d_{\mathrm{obs}} = 0$ $[m]$ is the horizontal position of the elliptic obstacle (from the stance foot), $w_{\mathrm{obs}} = 0.2$ $[m]$ and $h_{\mathrm{obs}} = 0.05$ $[m]$ are the width and height of the ellipse.

Optimization variables for this trajectory are $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, T, $\boldsymbol{u}$, r and $\bar{\kappa}$.

B. Stepping-down Trajectory

The stepping-down trajectory takes over where the walking trajectory left off at $t = T$ and does a stepping down motion while trying to keep the step length the same. The constraints for the stepping-down trajectory are set as follows (terms related to this trajectory are indicated with $\hat{}$ notation):

- Stepping-down trajectory continues from the end point of the walking trajectory: $\hat{\boldsymbol{x}}(0) = \boldsymbol{x}(T)$
- Swing foot related constraints: -0.11 [m]$\leq \hat{y}_{\mathrm{sw}}(\hat{T}) \leq -0.1$ [m], $\dot{\hat{y}}_{\mathrm{sw}}(\hat{T}) < -0.2$ [m/s], $\dot{\hat{x}}_{\mathrm{sw}}(\hat{T}) < 0$ [m/s]

This trajectory uses the same biarticular spring parameters obtained in "walking trajectory" optimization and variables are $\hat{\boldsymbol{x}}$, $\dot{\hat{\boldsymbol{x}}}$, $\hat{T}$, $\hat{\boldsymbol{u}}$. Here we set the step down height to be about 10 cm which can be adjusted depending on the task or terrain.

IV. CONTROL

In this section, we will introduce a controller that can handle blind walking on uneven terrain using the reference trajectories generated by direct collocation optimization. We will use a feedback linearization scheme for trajectory tracking. The model has 5 degrees of freedom but only 4 joints are controlled (point-foot model) resulting in underactuation. Because of this, we will formulate all the reference trajectories as a function

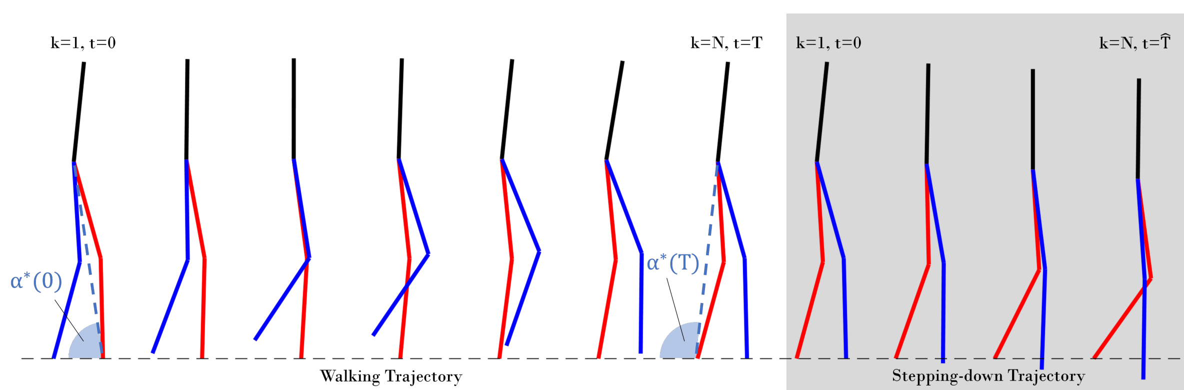

Fig. 5. Snap-shots of resulting walking trajectory and stepping-down trajectory where red links are the stance leg and blue links are the swing leg

of the virtual stance leg angle α as the phase variable (Fig. 1) to stabilize the system. Phase variable based implementations have been shown to be more robust compared to time based ones [2].

In the optimization part, α was constrained to be monotonically increasing. Reference trajectories are indicated by the * term where $\theta_i^*(\alpha)$ is the reference joint angle and $\dot{\theta}_i^*(\alpha)$ is the reference angular velocity for the i^{th} joint. Since direct collocation just outputs the results at collocation points, we linear interpolate for the in-between values.

The single stance phase ends when the swing foot contacts the ground (touch-down) and begins when it ceases contact with it (lift-off). In this paper, double stance phase is an instantaneous event as mentioned in Section II.

Reference walking trajectory obtained in Section III is defined in $\alpha \in [\alpha^*(0), \alpha^*(T)]$ where $\alpha^*(T)$ is the virtual stance leg angle at the end of the trajectory where swing foot contacts the ground. Using a predefined trajectory for walking on rough terrain is tricky because reference is defined for a certain range of α. If the α is within the defined range in the beginning of the single stance phase, reference walking trajectory can be used without any modifications. However, the single stance phase could begin with a lift-off virtual stance leg angle that is out of the defined range ($\alpha^{\text{LO}} < \alpha^*(0)$ or $\alpha^{\text{LO}} > \alpha^*(T)$) because of the random ground height. For the $\alpha^{\text{LO}} < \alpha^*(0)$ case, we use the modified α,

$$\begin{aligned} \hat{\alpha} &= \alpha + a\alpha + b \\ a &= \frac{\alpha^*(0) - \alpha^{\text{LO}}}{\alpha^{\text{LO}} - \alpha^{\text{merge}}}, \\ b &= -a\alpha^{\text{merge}} \end{aligned} \tag{10}$$

for obtaining reference trajectories $\theta_i^*(\hat{\alpha})$, $\dot{\theta}_i^*(\hat{\alpha})$. This makes sure α is in the defined range and linearly merges to the original one at α^{merge}. α^{merge} was chosen as the middle collocation point ($k = 12$). If $\alpha^{\text{LO}} > \alpha^*(T)$ at the beginning of the single stance phase, the gait is considered to have failed.

Another difficulty that can occur when walking on rough terrain is when the robot reaches the end of the reference walking trajectory but the expected touch-down condition ($y_{\text{sw}} \leq y_{\text{ground}}$) is not satisfied ($\alpha > \alpha^*(T)$). This is when the reference trajectory switches from "walking" to "stepping-down" and just tries to reach the ground with the swing foot while keeping the same step-length (and satisfying the other constraints).

TABLE I
5 LINK MODEL PARAMETERS

$l_1 = l_4 : 0.48$ [m]	$l_2 = l_3 : 0.48$ [m]	$l_5 : 0.48$ [m]
$m_1 = m_4 : 5$ [kg]	$m_2 = m_3 : 5$ [kg]	$m_5 : 60$ [kg]
$I_i = m_i l_i^2/12$ [kg $\cdot$ m^2], $i = 1, 2, 3, 4, 5$		

A diagram of the controller can be seen in Fig. 6. The reference trajectory generator sends the appropriate robot configuration as the reference according to the current α. We use feedback linearization to track actuated joint trajectories. Inputs can be chosen as:

$$\mathbf{u} = (\mathbf{T}\mathbf{M}^{-1}\mathbf{S})^{-1}(\boldsymbol{v} + \mathbf{T}\mathbf{M}^{-1}(\mathbf{H} - \boldsymbol{\tau})), \tag{11}$$

to linearize the system given in Equation (1) where $\mathbf{T} \in \mathbb{R}^{4\times5}$ is the task space matrix that maps the generalized coordinates to the actuated ones and

$$\boldsymbol{v} = \mathrm{K_p}\boldsymbol{y} + \mathrm{K_d}\dot{\boldsymbol{y}}, \tag{12}$$

$$\boldsymbol{y} = \begin{bmatrix} \theta_2^*(\alpha) - \theta_2 \\ \theta_3^*(\alpha) - \theta_3 \\ \theta_4^*(\alpha) - \theta_4 \\ \theta_5^*(\alpha) - \theta_5 \end{bmatrix}. \tag{13}$$

$\mathrm{K_p}$ and $\mathrm{K_d}$ are the proportional and derivative gains and are set to same values for each actuated joint.

V. RESULTS AND DISCUSSION

In this section, we will present the results when the trajectories obtained in Section III are used in the controller proposed in Section IV on the robot model described in Section II, where biarticular muscle parameters are set to those obtained by direct collocation optimization. We also investigate the effects of different biarticular muscle parameter combination's effect on robustness.

Simulations were performed in Matlab SIMULINK environment with variable step ode45 solver, max step size of 1e-3 and an absolute tolerance of 1e-8. The physical parameters of the robot are provided in Table I.

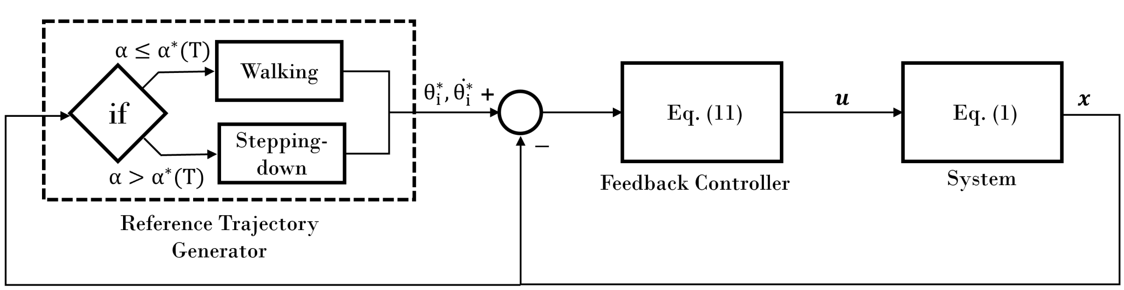

Fig. 6. The controller diagram

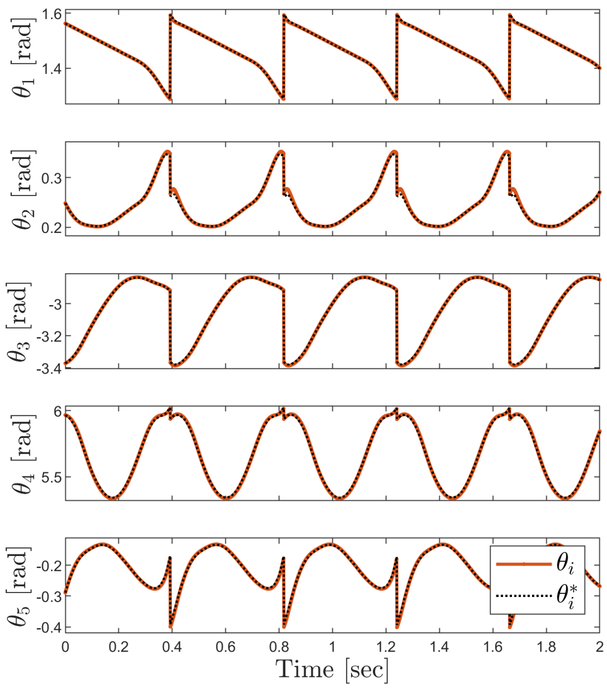

Fig. 7. Trajectory tracking results on flat terrain

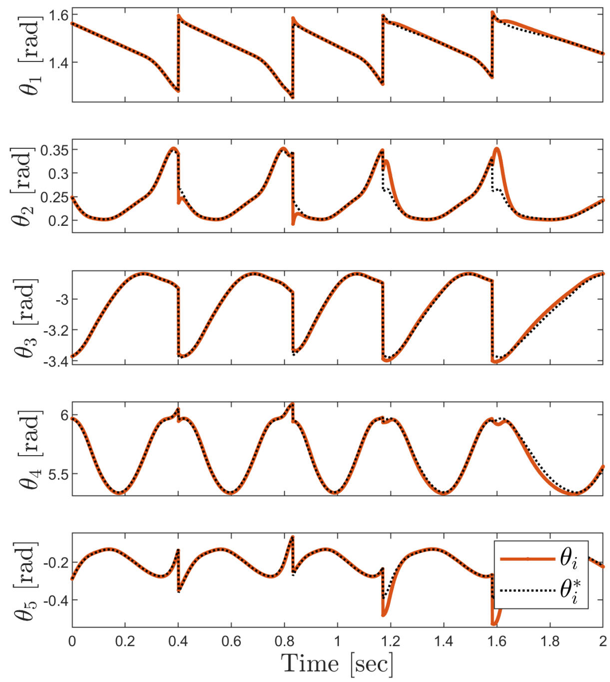

Fig. 9. Trajectory tracking results on rough terrain with $\delta = 0.05$ [m]

A. Walking on flat terrain

Fig. 7 shows the proposed controller's performance on a flat terrain. We can see that tracking performance is quite good for this under-actuated system. Gains were set to $K_p = 9700$ and $K_d = 220$ for this simulation. As mentioned in Section III, biarticular spring parameters were also optimized as they were also set as optimization parameters. We set them to resulting $r = 1.8173$ and $\bar{\kappa} = 29.8220$ [Nm] values for this gait. Average velocity was 0.61 [m/s].

B. Walking on rough terrain

Now, we will present the performance of the controller when the 5-link model is set to walk blindly (without any information of the terrain height changes) on the randomly generated terrains described in Section II-B. Gains and the biarticular spring parameters are kept the same as those mentioned in Section V-A. Terrain difficulty was set to $\delta = 0.05$ [m].

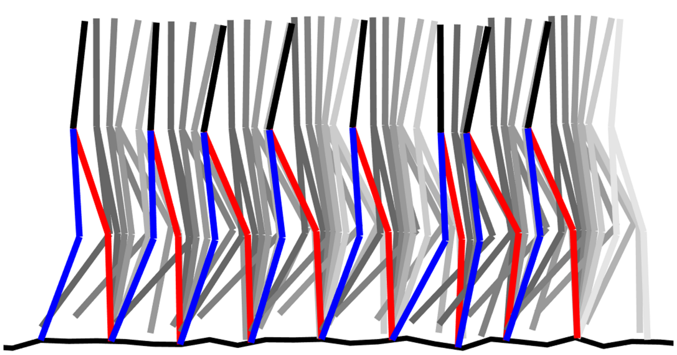

Fig. 8. Snap-shots from walking on a rough terrain with $\delta = 0.05$ [m] (Biarticular muscles were not shown on this figure to reduce visual clutter)

Fig. 8 shows some snap shots from this gait. Trajectory tracking performance of the proposed controller can be seen in Fig. 9. Trajectory tracking performance is still quite good considering the unknown terrain height changes the robot has to handle. It can be seen that the duration of a single step is not constant anymore and depends on the terrain. Fig. 10 shows the trajectory of the virtual stance leg angle during this gait. In this figure, it is easier to see that the duration of a step is not constant. Also, we can see that α can go out of the reference trajectory bounds (shown by the dotted lines). When α goes above the dotted line ($\alpha > \alpha^*(T)$), the controller switches the reference to stepping-down trajectory and when α starts below the dotted line, the modification described in Equation 10 is triggered. Overall, we can see that the proposed controller can handle walking on random rough terrain with good trajectory tracking performance. Average velocity was 0.442 [m/s] for this gait.

C. Effect of biarticular muscle parameters on rough terrain walking

We wanted to see the effect of different biarticular spring parameters on robustness, especially for the blind-walking on rough terrain case. In our model, biarticular springs have two parameters that can be adjusted: the lever arm ratio r and spring constant $\bar{\kappa}$.

Our metric of robustness is the maximum δ parameter the robot can handle, i.e. $\bar{\delta}$. We introduced how the random rough terrain was generated in Section II-B and terrain difficulty increased as δ parameter was increased. Fig. 11 shows the resulting $\bar{\delta}$ values in a 3D plot for combinations of $r = [1, 1.2, 1.4, ..., 5]$ and $\bar{\kappa} = [0, 5, 10, 15, ..., 200]$ [Nm]. For each combination, we start with $\delta = 0$ (flat terrain) and run the walking simulation. If the robot is successful at walking in the terrain for 10 seconds without falling, it is considered a successful walk and we increase δ by 0.001 [m] and run the simulation again. We do this until the robot can't handle the terrain difficulty anymore and the maximum δ value it can handle is recorded as $\bar{\delta}$ and is shown in the figure. Fig. 11 is the average result of simulation experiments in 10 different random terrain (4 of them are shown in Fig. 3).

In Fig. 11, $\bar{\kappa} = 0$ corresponds to a model without biarticular springs (we call this the default model) where $\boldsymbol{\tau} = 0$ in Equation (1). We can see that by adding biarticular springs, the robot can handle rougher terrains for some spring parameter combinations. The maximum terrain roughness that the default model could handle was $\delta = 0.0449$ [m]. It can be seen that there are a lot of r and $\bar{\kappa}$ combinations that can surpass this value and make the system more robust for blind-walking on rough terrain. The maximum value was reached with a parameter combination of $r = 2.2$, $\bar{\kappa} = 105$ [Nm], the robot was able to handle a terrain with $\delta = 0.0647$ [m] which is a 44.098% increase compared to the default model. Fig. 12 shows the same results for fixed r values in a 2D plot which is a bit more easier to read.

These are the average values for walking on different random terrains. For one of the terrains, robot was able to increase its $\bar{\delta}$ value from 0.0500 [m] for the default model to 0.0960 [m] by using biarticular muscles with $r = 2$, $\bar{\kappa} = 130$ [Nm] which is a 92% increase. The smallest maximum increase for a single terrain was from $\bar{\delta} = 0.0500$ [m] for the default model to $\bar{\delta} = 0.083$ [m] for the model with biarticular muscles which is a 66% increase. We can see that adding biarticular springs always ended up increasing the robustness for different random terrain. The reason for values for single terrains being larger than the average value is that different biarticular spring parameter settings perform better for different terrain.

However, we can also see that for larger r and $\bar{\kappa}$ values,

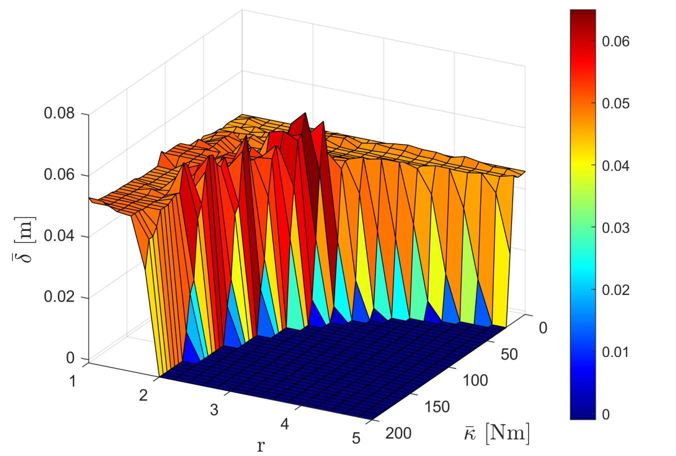

Fig. 11. Effects of different biarticular spring parameter combinations on robustness to terrain difficulty. Vertical axis shows $\bar{\delta}$ which is the maximum terrain difficulty the model with indicated parameter combination can handle. Larger $\bar{\delta}$ means it is more robust.

the robot can't even walk on even terrain (shown by the dark blue region in Fig. 11). This is because the biarticular muscles generate large torques that actuators can't handle(there is a $\pm$ 200 [Nm] torque limit on the actuators).

We have also investigated the efficiency of walking for different biarticular spring parameters that was used in this paper. Specific resistance (SR) [14]

$$SR := \frac{p}{Mgv}, \; p = \frac{1}{T}\int_0^T \sum_{i=2}^{5} |u_i \omega_i| dx, \tag{14}$$

was chosen as the efficiency indicator where T [secs] is the end time of one step, M [kg] is the total weight of the robot, g $[m/s^2]$ is the gravitational term, v $[m/s]$ is the average speed and p $[J/s]$ is the average input energy. A smaller SR value means that the gait is more energy-efficient. When

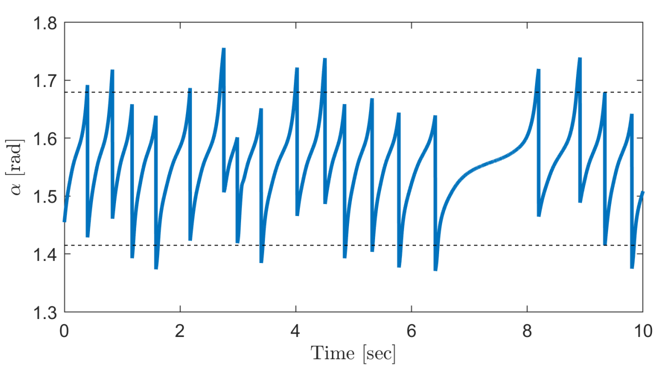

Fig. 10. Virtual stance leg angle α when walking on rough terrain with $\delta = 0.05$ [m]. The dotted lines show the upper and lower limits of the reference walking trajectory $\alpha^* \in [\alpha^*(0), \alpha^*(T)]$

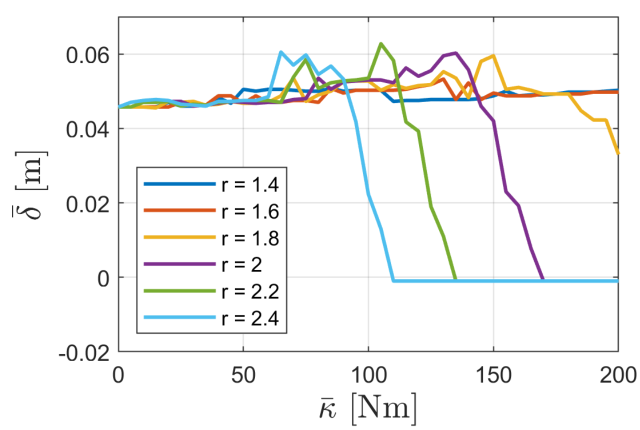

Fig. 12. This figure shows the same results presented in Fig. 11 but for fixed r values

walking on flat ground, $SR = 0.2731$ for the default model ($\bar{\kappa} = 0$), $SR = 0.1818$ for the BA model with spring parameters obtained from the optimization ($\bar{\kappa} = 29.8220$ [Nm], $r = 1.8173$) and $SR = 0.6449$ for the BA model with most robust spring parameters ($\bar{\kappa} = 105$ [Nm], $r = 2.2$) values were obtained. This shows that for different criterion, best pair of parameters are different. Biarticular muscles can increase the efficiency as shown in [11] but if we want to make the robot more robust, some efficiency must be sacrificed. Best way would be to adjust the parameters according to the environment and the task. For example, $\bar{\kappa}$ can be changed by introducing a stiffness adjustment mechanism as in [15] and r can be adjusted by a variable radius mechanism as in [16].

VI. CONCLUSIONS

In this paper, we proposed a controller that can use optimized trajectories to traverse through rough terrain without perception/sensing. To achieve this, we used a reference stepping-down trajectory in addition to the reference walking trajectory and also by modifying the phase variable (α) of the reference trajectories when necessary. We also showed how we have obtained these trajectories using direct collocation trajectory optimization. Through simulation experiments, it was shown that a 5-link underactuated biped robot model was able to handle random rough terrain with height changes up to 4.49 cm on average.

Using this controller, we also investigated the effects of passive biarticular muscles on robustness. It was shown that adding biarticular springs can significantly increase the performance for terrain-blind walking on rough terrain. Model with biarticular muscles was able to handle a terrain with 6.47 cm height change on average which is a significant increase compared to the default model. We also investigated how different biarticular spring parameters effect the robustness and found that adding biarticular muscles increased the robustness unless a really stiff spring was chosen or lever arm ratio was set too high. Our study shows that a spring constant of $\bar{\kappa} = 105$ [Nm] and lever arm ratio of $r = 2.2$ gave the best performance.

This study showed that biarticular springs in combination with our proposed controller can handle terrain-blind walking. Being able to handle terrain variations without perception/sensing would truly ease the burden on the high level controller, computation times would decrease and failures due to errors in perception could be mitigated.

As a follow-up to this work, we would like to achieve a velocity tracking scheme with our controller and work on improving the robustness. In [17], we showed that having variable stiffness on a SLIP model can significantly increase the robustness against external pushes. Now with this paper, we showed that different biarticular muscle parameters may fare better for different terrain and parameters that provide better robustness do not necessarily provide the best walking efficiency. Also in another future study, we would like to see if we can further improve the overall robustness and efficiency by having variable stiffness biarticular muscles and adjustable lever-arm ratio via an adjustment mechanism and an accompanying controller.

REFERENCES

[1] A. Werner, R. Lampariello and C. Ott, "Trajectory optimization for walking robots with series elastic actuators," 53rd IEEE Conference on Decision and Control, 2014, pp. 2964-2970, doi: 10.1109/CDC.2014.7039845.

[2] X. Da, O. Harib, R. Hartley, B. Griffin and J. W. Grizzle, "From 2D Design of Underactuated Bipedal Gaits to 3D Implementation: Walking With Speed Tracking," in IEEE Access, vol. 4, pp. 3469-3478, 2016, doi: 10.1109/ACCESS.2016.2582731.

[3] M. S. Jones, "Optimal control of an underactuated bipedal robot," M.S. thesis, 2014. [Online]. Available: http://ir.library.oregonstate.edu/xmlui/handle/1957/50690

[4] M. M. Pelit, J. Chang, R. Takano and M. Yamakita, "Bipedal Walking Based on Improved Spring Loaded Inverted Pendulum Model with Swing Leg (SLIP-SL)," 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2020, pp. 72-77, doi: 10.1109/AIM43001.2020.9158883.

[5] Y. Liu, P. M. Wensing, J. P. Schmiedeler and D. E. Orin, "Terrain-Blind Humanoid Walking Based on a 3-D Actuated Dual-SLIP Model," in IEEE Robotics and Automation Letters, vol. 1, no. 2, pp. 1073-1080, July 2016, doi: 10.1109/LRA.2016.2530160.

[6] G. Zhao, O. Mohseni, M. Murcia, A. Seyfarth, M. Sharbafi, "Exploring the effects of serial and parallel elasticity on a hopping robot," Frontiers in Neurorobotics, vol. 16, 2022, doi: 10.3389/fnbot.2022.919830.

[7] B. Fernini and M. Temmar, "The effect of mono and biarticular muscles on the dynamic of walking bipedal robot," 2017 Intelligent Systems Conference (IntelliSys), 2017, pp. 969-978, doi: 10.1109/IntelliSys.2017.8324247.

[8] M. Kumamoto, T. Oshima, T. Yamamoto,"Control properties induced by the existence of antagonistic pairs of bi-articular muscles — Mechanical engineering model analyses," Human Movement Science, Volume 13, Issue 5, 1994, pp. 611-634,, doi: 10.1016/0167-9457(94)90009-4.

[9] Jongwoo Lee, G. Lee, S. Hong, SangWook Lee, J. H. Kim and Y. Oh, "A novel multi-articular leg mechanism for biped robots inspired by bi-articular muscle," 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), 2016, pp. 1372-1377, doi: 10.1109/BIOROB.2016.7523825.

[10] K. Hosoda, Y. Sakaguchi, H. Takayama, and T. Takuma, "Pneumatic-driven jumping robot with anthropomorphic muscular skeleton structure," in *Autonomous Robots*, vol. 28, no. 3, pp. 307-316, 2010.

[11] M. M. Pelit, J. Chang and M. Yamakita, "Effects of Passive Biarticular Muscles on Walking Performance for Bipedal Robots," 2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2021, pp. 917-922, doi: 10.1109/AIM46487.2021.9517354.

[12] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review, Society for Industrial and AppliedMathematics*, vol. 59, no. 4, 2017, pp. 849–904.

[13] J. Koenemann, G. Licitra, M. Alp, M. Diehl, "OpenOCL - Open Optimal Control Library," in Robotics Science and Systems, workshop submission, June 2019.

[14] G. Gabrielli and T. H. von Karman, "What price speed?" Mechanical Engineering, vol. 72, no. 10, pp. 775–781, 1950.

[15] A. G. Rodríguez, J. M. Chacón, A. Donoso, A. G. G. Rodríguez, "Design of an adjustable-stiffness spring: Mathematical modeling and simulation, fabrication and experimental validation," Mechanism and Machine Theory, vol. 46, no. 12, 2011, pp. 1970-1979, doi: 10.1016/j.mechmachtheory.2011.07.002.

[16] X. Xiong, X. Sun, W. Chen, Y. Zhi and X. Fang, "Design of a Variable Stiffness Actuator Based on Variable Radius Mechanisms," 2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2022, pp. 1567-1572, doi: 10.1109/AIM52237.2022.9863246.

[17] M. M. Pelit and M. Yamakita, "Robust walking control based on the extended variable stiffness SLIP model," 2022 American Control Conference (ACC), 2022, pp. 538-543, doi: 10.23919/ACC53348.2022.9867311.