Design and Development of a One-Legged Hopping Robot Based on a Spring-Mass Template

Ahmet Safa Ozturk[1] and Ömer Morgül[1] and Ismail Uyanik[2]

***Abstract*— This work introduces a modular experimental platform for studying legged locomotion, constructed using the template-anchor methodology, which allows the generalization of designed applications to the broader spectrum of legged robots. A finite state machine governs the operation logic, enabling reliable phase-specific coordination within a hybrid dynamical system. Active hip torque control is integrated to stabilize trajectories and support continuous running, enhancing the dynamic capabilities of the platform. The experimental platform enables advanced system identification, forward prediction, and closed-loop control applications.**

I. Introduction

Developing and validating legged locomotion applications for complex tasks requires versatile experimental platforms [1]. Designing such platforms is often achieved with the template-anchor methodology by employing the spring-mass running template [2], [3]. Existing experimental platforms showed that this methodology is a promising development technique for legged robots [4], [5], [6].

This study contributes to this area by developing a modular and extensible experimental platform designed to investigate the generalization of template-based control strategies systematically. The experimental platform, designed using the spring-mass running template and implemented as a spring-mass planar hopper, serves as a testbed for exploring model-based and data-driven applications under realistic and hybrid dynamic conditions.

A finite state machine (FSM) governs the real-time operation framework, which manages the hybrid nature of the locomotion template with event-driven transitions. The FSM ensures the reliable coordination of the hybrid dynamics for robust and repeatable locomotion experiments.

The platform further incorporates active hip torque control, improving trajectory tracking and enhancing locomotion stability. It also supports developing advanced techniques, including forward prediction and control applications.

The remaining sections detail the underlying locomotion model (Sec. II), the experimental platform design (Sec. III), the real-time framework (Sec. IV), presents open-loop experimental results (Sec. V) and concludes (Sec. VI).

II. Extended Legged Locomotion Template

The spring-mass template elegantly models running with a massless springy leg attached to a point-mass body to analyze

[1]Ahmet Safa Ozturk and Ömer Morgül are with the Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Türkiye {sozturk,morgul}@ee.bilkent.edu.tr

[2]Ismail Uyanik is with the Department of Electrical and Electronics Engineering, Hacettepe University, 06800 Ankara, Türkiye uyanik@ee.hacettepe.edu.tr

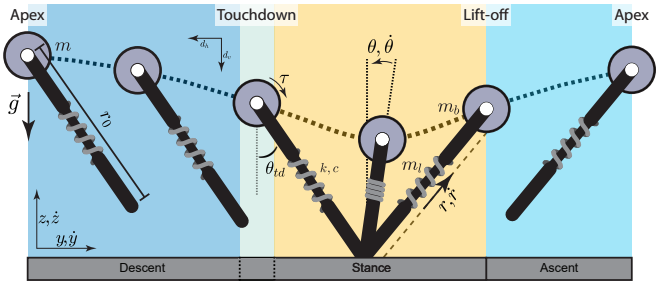

Fig. 1. Legged locomotion template with transition events and phases. The gray circle represents the body, and the black line with the spring represents the leg. The white dot at the center of the body also represents the hip. The touchdown event is not explicitly marked for the experimental setups, as the exact location of the touchdown is dependent on the precise ground profile.

Center of Mass (CoM) trajectories [3], [7]. Fig. 1 illustrates the template, with parameters defined in Table I.

The spring-mass template analyzes running movement as periodic strides, characterized using Poincaré sections chosen at the *apex* points where the body is at the maximum height with zero vertical velocity, $\dot{z}_a[n]$. The state vector at the nth *apex* is denoted as $\mathbf{X}_a[n] = [\dot{y}_a[n], z_a[n]]^T$ omitting cyclic $y_a[n]$. The *apex* triggers the stride with the *descent* phase, where the body (positioned at the **hip** (upper end of the leg)) follows a ballistic trajectory under gravity and air damping.

The *descent* phase concludes with the *touchdown* event, where the toe (lower end of the leg) contacts the ground with the touchdown angle, θ_{td}–the primary control input. The *touchdown* transitions the model into the *stance* phase, where the toe is stationary. Still, the body rotates due to the nonzero horizontal velocity and the hip torque, τ–the secondary control input. The body positions during *stance* are described

TABLE I
Parameter Definitions

Parameter	Definition
$y, \dot{y}, z, \dot{z}$	Horizontal and Vertical Body Positions and Velocities
$y_t, \dot{y}_t, z_t, \dot{z}_t$	Horizontal and Vertical Toe Positions and Velocities
$r, \dot{r}, \theta, \dot{\theta}$	Leg Length, Leg Angle and Velocities
r_0, m, m_b, m_l	Leg Rest Length, Total Mass, Body Mass, Leg Mass
k, c	Leg Spring Stiffness and Leg Damping Coefficients
d_h, d_v	Air Viscous Damping Factors (Horizontal and Vertical)
τ	Hip Torque
g, t	Gravitational Acceleration, Time

in Polar coordinates with r and θ as given in Fig. 1. The leg spring is also compressed and then decompressed during the *stance*, inverting the vertical velocity. The *stance* ends at the *lift-off* event where the toe clears from the ground and the motion enters the *ascent* phase (same dynamics as the *descent* apply). The *ascent* concludes at another *apex* $\mathbf{X}_a[n+1]$ marking the end of the stride [3], [7], [8], [9].

To complete the essential requirements of real-life applications, the base spring-mass template is extended by incorporating leg spring damping c and hip torque τ during stance, following the Torque-Actuated Dissipative model [9]. The extended template also includes air damping (d_h, d_v) and leg mass (m_l) effects during flight phases based on [10].

A. Extended Template Dynamics

1) Flight Phase: The flight dynamics are the ballistic motion of a point-mass given with $[\ddot{y}, \ddot{z}]^T = [-d_h\dot{y}, -d_v\dot{z} - g]^T$. The position and velocity trajectories are obtained by directly integrating the flight dynamics [10].

2) Stance Phase: The stance dynamics are derived with the Lagrange-Euler formulation, where the linear and angular momentum terms are defined as:

$$\frac{\mathrm{d}}{\mathrm{d}\,t}\begin{bmatrix} m_b\dot{r} \\ m_b r^2\dot{\theta} \end{bmatrix} = \begin{bmatrix} m_b r\dot{\theta}^2 - m_b g\cos\theta - k\Delta r - c\dot{r} \\ m_b g r\sin\theta + \tau \end{bmatrix}, \tag{1}$$

where the change in leg length is $\Delta r = r - r_0$. The above dynamics are non-integrable [11], so their solutions are generally obtained through alternative approximations [9].

As the above dynamics are only derived for the body, the effect of the leg mass on the movement can be modeled as an inelastic collision with the ground at the lift-off:

$$\left[\dot{y}^+, \dot{z}^+\right]^T = \left[\dot{y}^-, \dot{z}^-\right]^T \frac{m_b}{m_b + m_l}. \tag{2}$$

The body velocities immediately before and after the lift-off are denoted with superscripts $^-$ and $^+$, respectively [10].

B. Template-Driven Applications

1) Prediction Applications: This application aims to find an apex return map, $\mathcal{P}$, between the consecutive apexes $\mathbf{X}_a[n] \mapsto \mathbf{X}_a[n+1]$ with a known control input pair, $\mathbf{u}[n] : [\theta_{td}, \tau]$, such as [12]:

$$\mathbf{X}_a[n+1] = \mathcal{P}\left(\mathbf{X}_a[n], \mathbf{u}[n]\right). \tag{3}$$

Significant techniques for obtaining $\mathcal{P}$ include numerical integration or analytical approximation of the model dynamics [8], [9], [10]. $\mathcal{P}$ can also be obtained through implicit data-driven methods such as neural networks or Gaussian process regression (GPR) [13]. Prediction applications often require the identification of physical parameters, such as spring stiffness and damping coefficients, through an optimization of the experimental locomotion data.

2) Control Applications: The closed-loop control can be formulated as $\hat{\mathbf{u}}[n] = \mathcal{C}\left(\mathbf{X}_a[n], \mathbf{X}_a^d[n+1]\right)$ where a control input pair $\hat{\mathbf{u}}[n] = \left[\hat{\theta}_{td}, \hat{\tau}\right]^T$ is computed to convey the movement from the last observed apex, $\mathbf{X}_a[n]$, to a desired next apex, $\mathbf{X}_a^d[n+1]$ (See Remark 1). A common technique to obtain the control inputs is to optimize through a pre-developed forward predictor. This optimization can be done with any selected predictor, including numerical integration [13], analytical approximation [9], or data-driven approximation [13]. Another method for controller development is to define straightforward analytical relations between the state vectors and the control inputs [4].

Remark 1: The closed-loop controller requires the observed value of the apex, $\mathbf{X}_a[n]$, rather than computed or approximated values, $\hat{\mathbf{X}}_a[n]$, to regulate the desired locomotion behavior. Alternatively, an open-loop controller can be used with or without utilizing $\hat{\mathbf{X}}_a[n]$ to reduce the computational overheads to the experimental platforms.

III. Experimental Platform

The direct realization of the spring-mass template is challenging, as the motion is modeled on a plane. Planarizer mechanisms are prominently used to develop experimental applications based on the spring-mass hoppers, as they provide continuous running tracks without environmental measurements or external actuation. The planarizer mechanism features a single link, called the boom, pivoting at the origin on the ground with an unactuated R-R joint (two revolute joints). The robot is connected to the boom end and can move freely on a hemispherical surface [4].

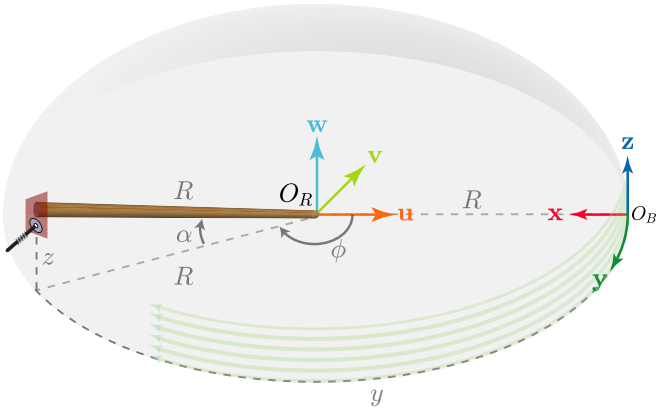

Fig. 2. Planarizer mechanism. The planarizer is depicted as the brown boom from the center to the left, and the leg assembly is shown with a maroon plate and the leg at the left end of the planarizer. The R-R joint, where the boom is connected, is at the origin of the planarizer reference frame O_R. The boom length is R, and the azimuth and elevation angles are ϕ and α, respectively. The spring-mass hopper is placed at the free end of the boom, where the origin of the body reference frame O_B is defined. The movement plane of the spring-mass hopper is depicted with the lime lines from center-right to center.

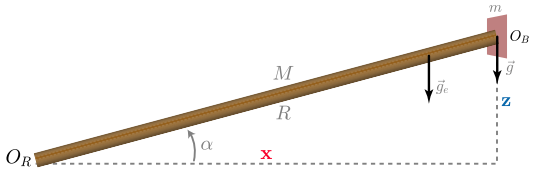

Fig. 3. Demonstration of the inertial effects of the boom mass M on the modified gravitational acceleration g. g_e is the gravitational acceleration of the Earth. As $R \gg z$, the small-angle approximation is used as $z = R\alpha$.

A. Planarizer Mechanism

The running template is modeled as a planar hopper, so using a sufficiently long boom concerning the body height trajectories, $R \gg z$, the workspace can be approximated as a cylindrical surface providing the required continuous movement plane of the spring-mass hopper. The planarizer is illustrated in Fig. 2. The length of the boom is $R = 1.34\,\mathrm{m}$. The experimental system is expressed in two reference frames: the planarizer reference frame, $\mathcal{F}_R = \{O_R, \mathbf{u}, \mathbf{v}, \mathbf{w}\}$, attached to the R-R joint where the boom is connected and the spring-mass body reference frame, $\mathcal{F}_B = \{O_B, \mathbf{x}, \mathbf{y}, \mathbf{z}\}$, which is attached at the free end of the boom and rotates with it. The $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ axes of $\mathcal{F}_B$ are defined as:

$$\mathbf{z} = \frac{\mathbf{u} \times \mathbf{v}}{\|\mathbf{u} \times \mathbf{v}\|}, \quad \mathbf{x} = -\frac{\mathbf{u}}{\|\mathbf{u}\|}, \quad \mathbf{y} = \mathbf{z} \times \mathbf{x}. \tag{4}$$

1) Planarizer Mechanics: Although the planarizer is not actuated, the inertial effects of the boom mass M on the movement are modeled as a modified gravitational acceleration, g, as depicted in Fig. 3 and given with [10]:

$$g \approx \frac{M/2 + m}{M/3 + m} g_e, \tag{5}$$

where g_e is the gravitational acceleration of the Earth. m and M are measured with precision scale and their values are $3.87\,\mathrm{kg}$ and $0.44\,\mathrm{kg}$ respectively. g_e at the geographical location of the experimental setup is $9.8\,\mathrm{m/s^2}$ (found with MATLAB `gravitywgs84` function), yielding the modified gravity, $g = 9.98\,\mathrm{m/s^2}$.

2) Planarizer Positions: The azimuth ϕ and the elevation α angles are measured with 3-channel optical encoders, each with 8192 counts per revolution resolution captured in quadrature mode. The encoders are connected with the boom with 1:6 timing belts, providing $360°/(8192 \times 6) = 0.0073°$ resolution in the angular measurements. The body positions and velocities from the boom angles are found as follows:

$$\begin{bmatrix} y[t] \\ z[t] \end{bmatrix} = R \begin{bmatrix} \phi[t] + 2\pi K \\ \sin \alpha[t] \end{bmatrix}, \tag{6a}$$

$$\begin{bmatrix} \dot{y}[t] \\ \dot{z}[t] \end{bmatrix} = \frac{1}{\Delta t} \begin{bmatrix} y[t] - y[t-1] \\ z[t] - z[t-1] \end{bmatrix}, \tag{6b}$$

where K is the number of positive turns around $\mathbf{w}$-axis and $\Delta t = 1\,\mathrm{ms}$ is the sampling period of the system. As numerical differentiation results in noisy velocity calculations, the velocities are alternatively filtered for precision-sensitive operations with $\left[\hat{\dot{y}}[t], \hat{\dot{z}}[t]\right]^T = [\mathcal{K}_h(y[0:t]), \mathcal{K}_v(z[0:t])]^T$, where $\mathcal{K}_h$ and $\mathcal{K}_v$ denote Kalman-like filter operators on the horizontal and vertical positions.

B. Leg Assembly

The leg assembly is attached to an end plate at the free end of the boom. The end plate is further connected to the central assembly with additional supporting booms. The connection between the end plate and the main and supporting booms is achieved through the freely rotating bearings, ensuring that the end plate is always perpendicular to the $\mathbf{uv}$-plane (ground). This constraint further ensures the reliability of the cylindrical plane approximation of the planarizer workspace. The leg assembly includes the hip actuation unit and the leg.

1) Hip Actuation: The hip is actuated with an Electronically Commuted (EC) motor. EC motors provide torque efficiency and durability as they do not contain rotor windings. Maxon EC40-393024 (170 W) brushless motor is selected as the hip actuator. A planetary gearhead (Maxon GP-42-C) with 1:26 reduction is also attached at the front of the motor to reduce the leg angular speed and increase the hip torque. The closed-loop position control of the hip motor is achieved with two different sensor measurements, including the 3-channel hall sensor, which provides sparse readings of the rotor positions to improve the power efficiency, and a 3-channel optical encoder, which provides precise measurements of the motor shaft positions to advance the position accuracy at low speeds. The resolution of the shaft encoder is 2000 counts per revolution in quadrature mode. Combined with the gear ratio, the resulting effective precision on the leg angle measurements is $360°/(2000 \times 26) = 0.0069°$.

The motor is used with three operation modes: current, angle, and angular velocity, where I_D, θ_D, and $\dot{\theta}_D$ denote

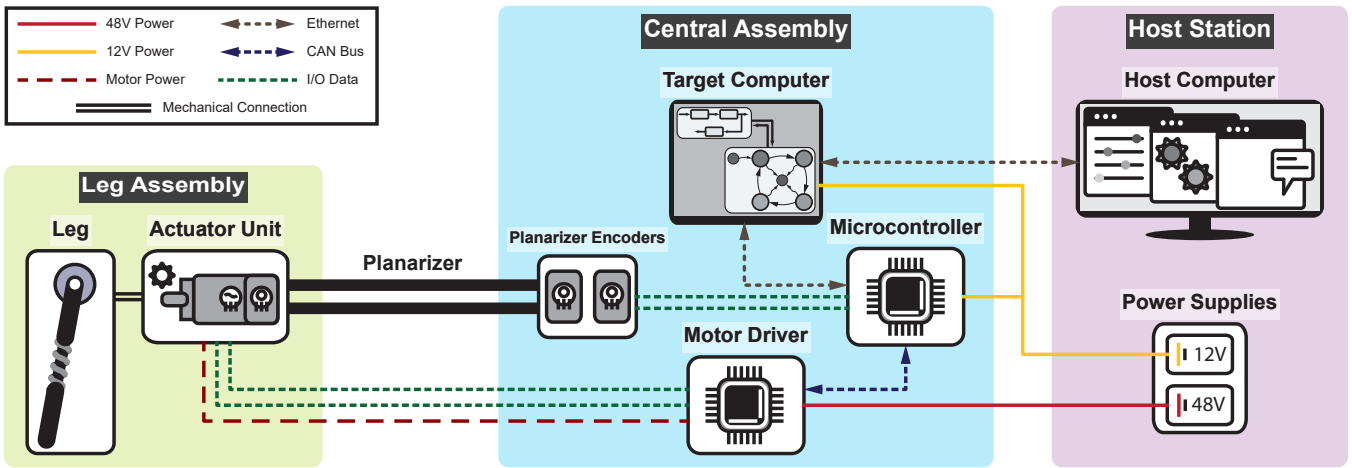

Fig. 4. Hardware Setup of the Experimental Platform.

the demand values of the actuation modes. The angle mode is used to set the touchdown leg angle, and the current mode is utilized to apply the hip torque. The angular velocity mode is also used to smooth the setting of the leg angle. The torque demand value, τ_D, is scaled to the current demand value with the torque factor, C_τ, as $I_D = \tau_D C_\tau$.

2) Leg Structure: The leg is designed to house the leg spring, allowing compression and decompression along the leg length, r-axis, with guide shaft and stoppers. The rest length of the leg is $r_0 = 21\,\text{cm}$. The leg is attached to the shaft of the planetary gear. The state vector of the toe is denoted as $\mathbf{X}_t[t] = [y_t[t], \dot{y}_t[t], z_t[t], \dot{z}_t[t]]^T$ and positions and velocities are obtained as:

$$\begin{bmatrix} y_t[t] \\ z_t[t] \end{bmatrix} = \begin{bmatrix} y[t] \\ z[t] \end{bmatrix} + r_0 \begin{bmatrix} \sin\theta[t] \\ -\cos\theta[t] \end{bmatrix}, \tag{7a}$$

$$\begin{bmatrix} \dot{y}_t[t] \\ \dot{z}_t[t] \end{bmatrix} = \frac{1}{\Delta t} \begin{bmatrix} y_t[t] - y_t[t-1] \\ z_t[t] - z_t[t-1] \end{bmatrix}. \tag{7b}$$

Note that the toe positions are only obtained for the flight phases where $r(t) = r_0$. Obtaining toe positions during stance is redundant, as the toe is stationary.

C. Central Assembly

The motor control, communication, and computation hardware are placed at the central assembly, which can rotate around $\mathbf{w}$-axis with the planarizer. The communication and power delivery with the central assembly is achieved through rotary slip rings. The hardware setup of the experimental platform is shown in Fig. 4.

The motor commands are relayed to the motor driver through a CAN bus connection from a microcontroller. As EC motors require specialized driver hardware, the commands are converted into phase signals before being applied to the motor. The motor driver is the Maxon EPOS2 Positioning Controller with 70V/10A capacity.

The communication between the motor driver and the computation hardware is handled by a separate microcontroller that also captures the planarizer encoder readings. Texas Instruments Hercules Microcontroller, which has a Quadrature Encoder, CAN Bus, and Ethernet modules, is used for the above operations. The Ethernet connection allows the connection between the microcontroller and the Target Computer, which is the primary computing hardware.

The Target Computer manages all operation framework tasks described in the following section in real-time with the Simulink Real-Time OS. The real-time operation frequency of the experimental platform is $1\,\text{kHz}$. Additionally, the Target Computer allows the inspection and configuration of the system through another Ethernet connection with the Host Computer. Providing reliability and durability, the E243739 Industrial Computer is selected as the Target Computer.

D. Host Station

The Host Station comprises the Host Computer and Power Supplies. The Host Computer allows tuning the motor driver and microcontroller through specialized software such as EPOS Studio and Code Composer Studio. The configuration for the Target Computer is also set through the Host Computer with MATLAB and Simulink. Finally, the Host Computer features two user interfaces, including a C#-based Visual Studio and Simulink-based MATLAB interfaces for online monitoring and inspecting the experimental platform.

IV. Real-Time Operation Framework

The experimental platform is operated through a finite state machine (FSM). The operation framework is deployed into phases where the transitions are handled using event-based logic. The functionality of the operation framework is detailed in Algorithm 1 and illustrated in 5.

The legged locomotion template given in Sec. II is further extended by the subdivision of the stance phase into *pre-stance* and *stance* phases. This subdivision adjusts the intended locomotion template into the physical system by allowing smoother running behavior, particularly by avoiding backjumps caused by uncertain touchdown detection. The transition into the *pre-stance* phase is triggered by the *pre-touchdown* event, detected as $z_t \leq z_t^{\triangleleft}$ where $z_t^{\triangleleft}$ is the clearance threshold above the ground which ensures the hip motor is turned off before the *stance* phase to guarantee the unconstrained motion of the body at the touchdown point without any hip motor resistance. While turning off the hip motor slightly perturbs the intended touchdown angle, maintaining the leg angle command until touchdown often results in undesired rebounds in the negative horizontal direction. After the *pre-stance* phase, the *touchdown* is detected when the toe reaches the ground $z_t \leq 0$.

The most computationally heavy component of the operation framework occurs during the *descent* phase. Depending on the user's preference, the control inputs can be applied either in open-loop or closed-loop mode. The control inputs can be calculated beforehand and provided directly to the framework in open-loop mode. In contrast, the closed-loop control mode requires the state vector of the last apex to compute control inputs for the desired trajectory. This computation must be completed during the *descent* phase as the touchdown angle of the leg must be set before the

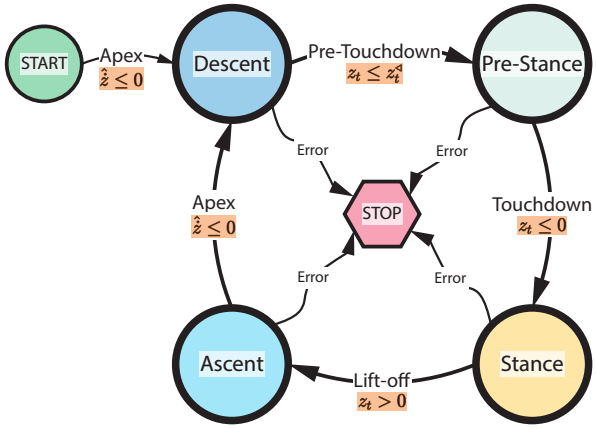

Fig. 5. FSM of the Operation Framework with Event Conditions.

Algorithm 1 Phase-Specific Framework Deployment with Transition Event Detection. $\breve{\mathbf{u}}$, $\hat{\mathbf{u}}$, and $\bar{\mathbf{u}}$ denote open-loop, closed-loop, and finalized control inputs, respectively. Superscripts $^{\triangleleft}$ and $^{\triangleright}$ indicate the lower and upper bounds.

Require: $\breve{\mathbf{u}}$ ▷ Open-Loop Control Inputs **or** $\mathbf{X}_a$, $\mathbf{X}_a^d$ ▷ Closed-Loop Positions
Require: $\hat{\dot{z}}$ ▷ Filtered Vertical Velocity of the Body (Hip)
Require: $\tau^{\triangleleft}$, $\tau^{\triangleright}$, C_τ ▷ Hip Torque Limits, Torque Factor
Require: z_t, $z_t^{\triangleleft}$ ▷ Toe Height and Clearance Threshold
Require: θ, $\dot{\theta}_A$ ▷ Leg Angle and Angular Ascent Velocity

```
switch Phase do
    case Start
        if ẑ̂ ≤ 0 then
            Phase←Descent                          ▷ First Apex
    case Descent
        if Open-Loop Mode then
            ū[n] ← [θ̆_td[n], τ̆[n]]^T
        else if Closed-Loop Mode then
            û[n] ← C(X_a[n], X_a^d[n+1])
            ū[n] ← û[n]
        τ̄[n] ← clamp(τ̄[n], τ^◁, τ^▷)               ▷ Limit Torque
        θ_D ← θ̄_td[n]                              ▷ Set Leg Angle
        if z_t ≤ z_t^◁ then                        ▷ Check for Toe Clearance
            Phase←Pre-Stance                       ▷ Pre-Touchdown
    case Pre-Stance
        I_D ← 0                                    ▷ Disable Hip Motor
        if z_t ≤ 0 then
            Phase ←Stance                          ▷ Touchdown
    case Stance
        if θ ≤ 0 then
            I_D ← τ̄[n]C_τ                          ▷ Apply Hip Torque
        if z_t > 0 then
            Phase←Ascent                           ▷ Lift-off
    case Ascent
        if z_t ≥ z_t^◁ then                        ▷ Check for Toe Clearance
            θ̇_D ← θ̇_A                              ▷ Rotate Leg
        if ẑ̂ ≤ 0 then
            Phase←Descent                          ▷ Apex
```

pre-stance phase. The hip torque also must be computed before *stance* phase, which is the only part of the locomotion to interact with the spring-mass dynamics. The hip torque actuation is applied only when $\theta \leq 0$, to maximize the energy injection to the system.

Additionally, the leg is actuated with a positive angular velocity after a suitable point during the *ascent* phase. This rotation allows a smoother leg angle change from a negative lift-off angle to a positive touchdown angle.

V. Experimental Results

A series of open-loop experiments is conducted to validate the functionality of the experimental platform and real-time framework with the range of control inputs: $\theta_{td} \in \{10^\circ, 15^\circ, 20^\circ, 25^\circ\}$ and $\tau \in \{4, 5, 6, 7\}\,\mathrm{N\,m}$. The locomotion data is collected using the detailed hardware in Sec. III and the framework given in Sec. IV running at $1\,\mathrm{kHz}$.

To show that the experimental platform is working as intended and the locomotion behavior can be analyzed using the platform, a continuous run from a selected experiment ($\theta_{td} = 15^\circ$, $\tau = 4\,\mathrm{N\,m}$) is given in Fig. 6.

Fig. 6 demonstrates the realization of experimental multi-stride locomotion by showing that the experimental platform is suitable for the intended locomotion behavior, and the FSM given in Algorithm 1 works as designed to detect the transition of stride phases.

The consistent $1\,\mathrm{ms}$ sampling intervals of the time-series data from Fig. 6 logged by the Simulink Real-Time OS, confirm the real-time operation frequency of $1\,\mathrm{kHz}$.

To better illustrate the motor commands applied during the FSM phases, the highlighted stride from Fig. 6 is further analyzed in Fig. 7, demonstrating the angle, θ_D, and current, I_D, command values sent to the hip motor along with the actual measurements of leg angle, θ, and current, I. It can be seen that the motor commands of the operation framework are set correctly to comply with the locomotion template, and their actual values show that the command values are tracked precisely by the system.

To better illustrate the open-loop stability characteristic of the system, the apex return maps under the different hip torque values ($\tau \in \{4, 5, 6, 7\}\,\mathrm{N\,m}$) with fixed touchdown angle, $\theta_{td} = 15^\circ$, is given in Fig. 8. Although the sustained multi-stride locomotion was demonstrated with the open-loop control of the experimental platform, Fig. 8 clearly shows that the apex return maps diverge for all of the reported cases. The instability of apex return maps is indeed an expected result since the open-loop controller is highly sensitive to user intervention, as the manual launch of the leg assembly initiates locomotion. These results motivate the use of closed-loop control for the experimental platform to stabilize locomotion behavior.

The presented experimental results show that the developed experimental platform and the designed operation framework are promising tools for developing and testing legged locomotion applications, which are generalizable to various legged robot morphologies, as the platform directly works with the legged locomotion template.

VI. Conclusion

This paper presented the design, implementation, and validation of a real-time operated experimental platform for legged locomotion. Open-loop experiments demonstrated multi-stride locomotion but highlighted inherent instability, motivating the need for feedback control. Future work will focus on implementing and testing model-based and data-driven closed-loop controllers to achieve stable locomotion.

References

[1] Z. Wu, K. Zheng, Z. Ding, and H. Gao, "A survey on legged robots: Advances, technologies and applications," *Eng. Appl. Artif. Intell.*, vol. 138, Dec. 2024. Art no. 109418.

[2] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *J. Exp. Biol.*, vol. 202, pp. 3325–3332, Dec. 1999.

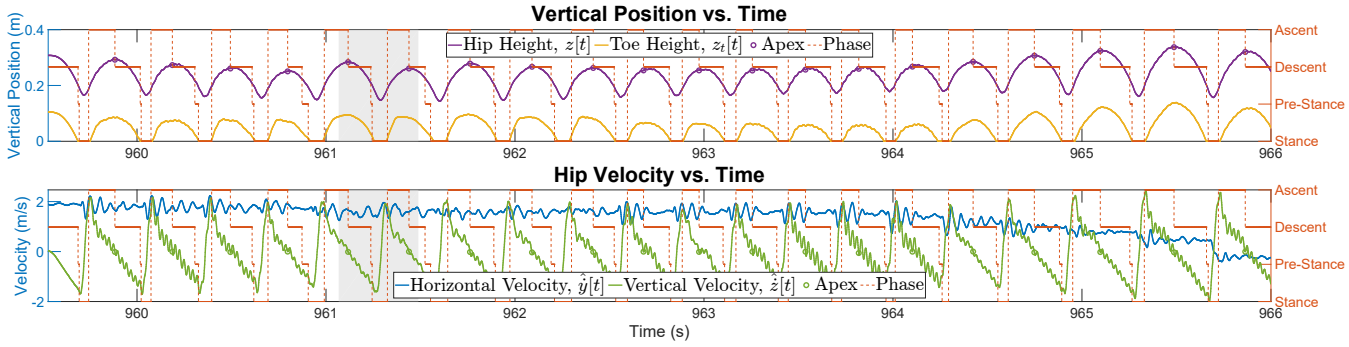

Fig. 6. Time Series Plot for an Example Run with an Open-Loop Controller ($\theta_{td} = 15°$, $\tau = 4\,\mathrm{N\,m}$). On top, the vertical positions of the hip and toe, and on the bottom, the filtered horizontal and vertical velocities of the hip are given. The apex detections are marked with o, and the stride phases are indicated with the step plot on the right y-axis. The shaded area between 961.067 s and 961.489 s is analyzed in Fig. 7 to investigate the motor commands applied to the system.

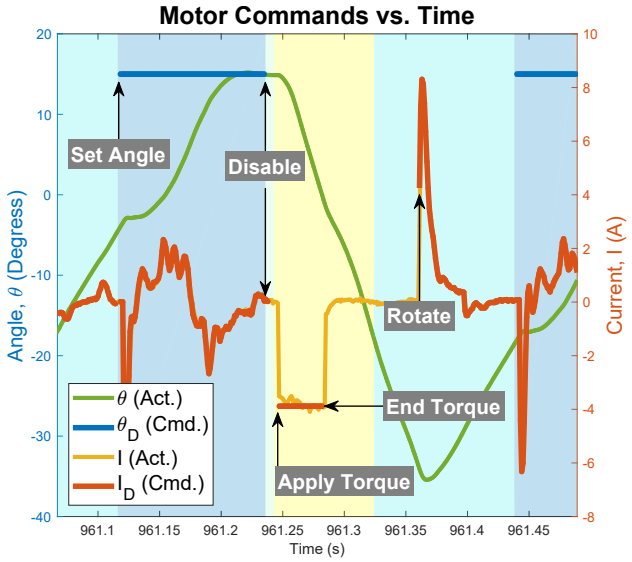

Fig. 7. Motor Commands During the Shaded Region of Fig. 6. Background colors indicate the phases as given in Fig. 1. 'Act.' and 'Cmd.' are the Actual and Commanded values of angle and current inputs to the motor. Command values are plotted only for the existing periods. Annotations indicate the key actions during the FSM given in Algorithm 1: touchdown leg angle is set during *Descent*, the hip motor is disabled for *Pre-Stance*, the hip torque is applied during *Stance* when $\theta \leq 0$, and the leg is rotated during *Ascent*.

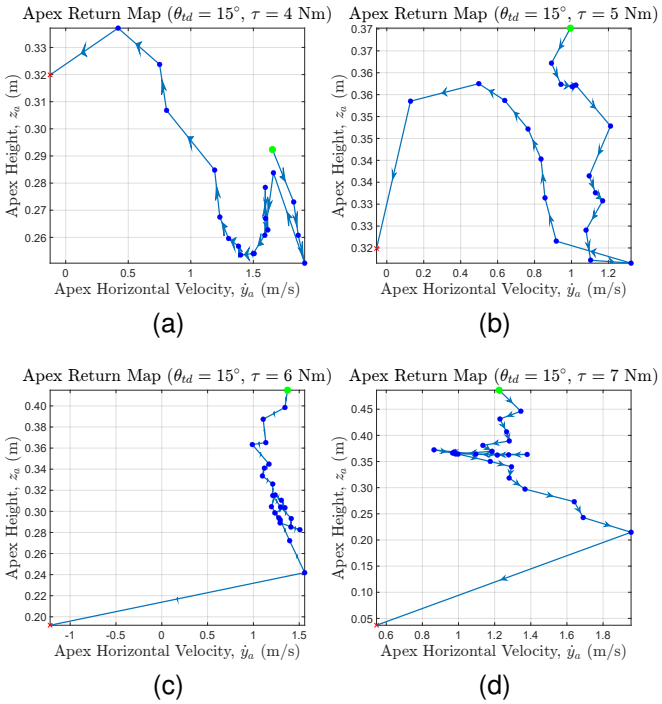

Fig. 8. Apex Return Maps of the Open-Loop Control Experiments. The initial and final apexes are marked with a green circle and a red cross, respectively. Blue circles mark the intermediate apexes, and the transitions are indicated with arrows. (a) $\theta_{td} = 15°$, $\tau = 4\,\mathrm{N\,m}$. (b) $\theta_{td} = 15°$, $\tau = 5\,\mathrm{N\,m}$. (c) $\theta_{td} = 15°$, $\tau = 6\,\mathrm{N\,m}$. (d) $\theta_{td} = 15°$, $\tau = 7\,\mathrm{N\,m}$.

[3] R. Blickhan, "The spring-mass model for running and hopping," *J. Biomechanics*, vol. 22, no. 11, pp. 1217–1227, 1989.

[4] M. H. Raibert, *Legged Robots That Balance*. Cambridge, MA, USA: MIT press, 1986.

[5] J. Oehlke, M. A. Sharbafi, P. Beckerle, and A. Seyfarth, "Template-based hopping control of a bio-inspired segmented robotic leg," in *2016 6th IEEE Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, pp. 35–40, 2016.

[6] A. De, T. T. Topping, J. D. Caporale, and D. E. Koditschek, "Mode-reactive template-based control in planar legged robots," *IEEE Access*, vol. 10, pp. 16010–16027, 2022.

[7] T. A. McMahon and G. C. Cheng, "The mechanics of running: How does stiffness couple with speed?," *J. Biomechanics*, vol. 23, pp. 65–78, 1990.

[8] H. Geyer, A. Seyfarth, and R. Blickhan, "Spring-mass running: simple approximate solution and application to gait stability," *J. Theor. Biol.*, vol. 232, pp. 315–328, Feb. 2005.

[9] M. M. Ankarali and U. Saranli, "Stride-to-stride energy regulation for robust self-stability of a torque-actuated dissipative spring-mass hopper," *Chaos: Interdisciplinary J. Nonlinear Sci.*, vol. 20, Sep. 2010. Art. no. 033121.

[10] I. Uyanik, Ö. Morgül, and U. Saranli, "Experimental validation of a feed-forward predictor for the spring-loaded inverted pendulum template," *IEEE Trans. Robot.*, vol. 31, pp. 208–216, Feb. 2015.

[11] P. Holmes, "Poincaré, celestial mechanics, dynamical-systems theory and "chaos"," *Phys. Rep.*, vol. 193, pp. 137–163, Sep. 1990.

[12] W. J. Schwind, *Spring loaded inverted pendulum running: A plant model*. PhD thesis, Dept. Elect. Eng., Univ. Michigan, Ann Arbor, USA, 1998.

[13] P. A. Bhounsule, M. Kim, and A. Alaeddini, "Approximation of the step-to-step dynamics enables computationally efficient and fast optimal control of legged robots," in *44th Mechanisms Robot. Conf. (MR)*, vol. 10 of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Aug. 17–19 2020. Paper V010T10A050.