Contact-Implicit Optimization of Locomotion Trajectories for a Quadrupedal Microrobot

Neel Doshi\textsuperscript{1,2}, Kaushik Jayaram\textsuperscript{1,2}, Benjamin Goldberg\textsuperscript{1,2}, Zachary Manchester\textsuperscript{3}, Robert J. Wood\textsuperscript{1,2}, and Scott Kuindersma\textsuperscript{1}

\textsuperscript{1}John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, M.A
Email: ndoshi@g.harvard.edu

\textsuperscript{2}Wyss Institute for Biologically Inspired Engineering, Harvard University, Cambridge, M.A.

\textsuperscript{3}Department of Aeronautics and Astronautics, Stanford University, Stanford, C.A.

Abstract—Planning locomotion strategies for legged microrobots is challenging due to their complex morphology, high frequency passive dynamics, and discontinuous contact interactions with the environment. Consequently, such research is often driven by time-consuming experimental tuning of controllers designed with simplified models. As an alternative, we present a framework for systematically modeling, planning, and controlling legged microrobots. We develop a three-dimensional dynamic model of a 1.43 g quadrupedal microrobot that has complexity (e.g., number of actuated degrees-of-freedom) similar to larger-scale legged robots. We then adapt a recent variational contact-implicit trajectory optimization method to generate feasible whole-body locomotion plans for this robot. We demonstrate that these plans can be tracked with simple joint-space controllers that are suitable for computationally constrained microrobots. Our method is used to plan periodic gaits at multiple stride frequencies and on various surfaces. These gaits achieve high per-cycle velocities, including a maximum of 10.87 mm/cycle, which is 33\% faster than previously measured for this robot. Furthermore, we plan and execute a vertical jump of 9.96 mm, which is 78\% of the robot’s body height. To the best of our knowledge, this is the first end-to-end demonstration of planning and tracking whole-body dynamic locomotion on a millimeter-scale legged robot.

I. INTRODUCTION

A. Motivation

Laminate manufacturing processes, such as smart-composite microstructures (SCM) \cite{40} or printed circuit MEMS \cite{37}, enable rapid and reliable assembly of flexure-based, millimeter-scale devices. A major advantage of these manufacturing techniques is the ability to realize mechanically complex devices with many degrees-of-freedom (DOFs) at the millimeter-scale. This has enabled the development of legged microrobots that do not sacrifice dexterity or mobility for (decreased) scale. These legged microrobots leverage favorable inertial scaling \cite{36} to demonstrate remarkable capabilities, including high-speed running \cite{12}, jumping \cite{13}, and climbing \cite{2}. However, these results have largely been achieved using simplified models \cite{14} and time-consuming experimental methods \cite{10} due to the mechanical complexity of the robots and challenges associated with modeling legged systems.

The ability to effectively model and control legged microrobots would greatly benefit work in this area. We evaluate a model-based optimization method for designing agile locomotion behaviors involving contact that avoids the need for exhaustive experimentation. These model-based tools are increasingly important as microrobots move closer to envisioned applications, including inspection in confined environments \cite{18}, search and rescue, and environmental monitoring. This work is also broadly applicable to other systems that interact with the environment through contact, including manipulators and larger legged robots.

B. Related Work

A variety of sophisticated model-based methods have been developed to design trajectories for legged robots, including hybrid \cite{29, 7, 31, 24, 30} and contact-implicit \cite{28, 34, 25, 22} trajectory optimization methods. Contact-implicit methods have the benefit that they generate contact sequences as part of the optimization, thereby eliminating the need for \textit{a-priori} contact mode scheduling. This enables planning for a variety of periodic and aperiodic behaviors. Most methods, however, are limited to first-order integration accuracy, which creates a linear tradeoff between the size of the trajectory optimization problem and the resulting trajectory accuracy. For high-dimensional robots with multiple contacts, this can lead to nonlinear programs with tens of thousands of variables that push the limits of modern solvers. In practice, coarse time discretizations are used to reduce program size, resulting in inaccurate whole-body locomotion plans that are difficult or impossible to realize on a physical robot.

Inspite of this drawback, contact-implicit methods are often used to plan for reduced dynamical models, such as centroidal dynamics. These low-dimensional trajectories are then stabilized using inverse dynamics controllers that resolve whole-body motions online \cite{32}. Such methods have achieved

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Still frames of a quadrupedal microrobot executing a vertical jump.}
\end{figure}
success on the humanoid Atlas robot [20], and the quadrupedal HyQ robot [38, 39]. Another approach, also implemented on the HyQ robot, is to use relaxed or spring-damper contact models specifically calibrated for a given system [21]. Despite these successes, whole-body motion plans produced using general physics-based models have yet to be realized on a physical robot.

Recently, Manchester and Kuindersma [22] developed a variational contact-implicit trajectory optimization scheme that combines ideas from discrete variational mechanics with the complementarity formulation of rigid-body contact to achieve higher-order integration accuracy for a particular trajectory optimization problem. Simulation results from that work demonstrate that this approach results in more accurate whole-body locomotion plans that can be tracked with simple controllers. In this paper, we extend and apply these methods to plan and track locomotion trajectories on a physical microrobot.

C. Contributions

Our primary contribution is the development and evaluation of a framework for modeling, planning, and controlling dynamic behaviors for legged microrobots. We develop a full three-dimensional dynamic model of a quadrupedal micro-robot with eight control inputs, a 76 dimensional state, and 24 kinematic position constraints. We also adapt a state-of-the-art variational contact-implicit trajectory optimization algorithm to generate physically accurate locomotion plans. Finally, we develop a low-latency estimator and joint-space controller that allow this robot to track the generated plans.

Our methods are used to generate nine periodic gaits at three frequencies (2, 10, and 30 Hz), on three different surfaces (sandpaper, card-stock, and Teflon). Optimized gaits for each condition are shown to move 17% faster per-cycle than heuristically tuned gaits at the same operating conditions. We also execute a gait with average velocity of 10.87 mm/cycle – the fastest recorded for this platform. Finally, we demonstrate the first controlled vertical jump of 9.6 mm, which is 78% of the quadrupedal microrobot’s body height.

D. Paper Organization

The remainder of this paper is organized as follows: we present an overview of the platform and a dynamic model of the system in Sec. [II]. The trajectory optimization problem for periodic and aperiodic behaviors is formalized in Sec. [III]. In Sec. [IV], we describe the hardware and software used for the locomotion experiments, and we present and discuss the results of these experiments in Sec. [V] and Sec. [VI] respectively. Finally, we draw conclusions and present directions for future research in Sec. [VII].

II. DYNAMIC MODEL

A. Platform Overview

The quadrupedal microrobot (Fig. 1) is 4.51 cm long, weighs 1.43 g, and has eight independently actuated DOFs. Each leg has two DOFs that are driven by optimal energy density piezoelectric bending actuators (henceforth actuators) [16]. These actuators are controlled with AC voltage signals using a simultaneous drive configuration described by Karpelson et al. [19]. A spherical-five-bar (SFB) transmission (Fig. 2) connects the two actuators to a single leg in a nominally decoupled manner: the swing actuator controls leg-x motion, and the lift actuator controls the leg-z motion. Each SFB transmission has 11 carbon fiber linkages (QA-112, Tohotenax) connected by nine compliant polyimide flexures (Kapton, Dupont) in three parallel kinematic chains.

B. Robot Model

The dynamics of the SFB transmissions are assumed to follow the pseudo-rigid body approximation [15], with the flexures and carbon fiber linkages modeled as pin joints and rigid bodies, respectively. Each flexure is assumed to deflect only in pure bending with its mechanical properties described by a torsional spring and damper that are sized according to the procedure described by Doshi et al. [5]. Given these assumptions, an intuitive dynamical description of each SFB transmission has two inputs (forces generated by the actuators), and eight generalized coordinates: two independent coordinates (actuator tip deflections) and six dependent coordinates (a subset of flexure joint angles). The remaining three flexure joint angles are used to define six constraints that represent the kinematics of the three parallel chains. Thus, a complete model of the robot has eight inputs, 38 generalized coordinates (76 states), and 24 position constraints.

A three-dimensional computer-aided-design model is developed using SolidWorks (Dassault Systèmes) to capture the kinematics and inertial properties of the microrobot. An open-source SolidWorks-to-Universal-Robot-Description-Format (URDF) exporter is used to generate an initial URDF model of the robot. Actuator forces, joint limits, kinematic-loop constraints, and the mechanics (stiffness and damping) of the flexural joints are then manually incorporated. Furthermore, units are rescaled from SI (seconds, meters, and kilograms) to milliseconds, millimeters, and grams for improved
numerical conditioning. The dynamics and control toolbox Drake \cite{drake} is used to compute the terms in the Euler-Lagrange equation from this URDF description using the composite rigid body algorithm \cite{pellegrino}. 

\[
\frac{d}{dt} D_2 \mathcal{L}(q, \dot{q}) - D_1 \mathcal{L}(q, \dot{q}) + C(q)^T \lambda = F^b(q, \dot{q}) + F^{act}(q) \\
C(q) = 0. \tag{1}
\]

Here $\mathcal{L}$ is the microrobot’s Lagrangian (including flexural spring energy), $q \in \mathbb{R}^{38}$ is the vector of generalized coordinates, $C(q)^T = (\partial c/\partial q)^T$ is the Jacobian mapping constraint forces, $\lambda$, into generalized coordinates, and the dot superscripts represent time derivatives. Note that the slot derivative $D_i$ indicates partial differentiation with respect to a function’s $i^{th}$ argument. $F^b$ is the vector of generalized flexural damping forces, and $F^{act}$ is the vector of generalized actuator forces. Equation (2) enforces the kinematic-loop constraints $c(q)$.

Each actuator is modeled as a force source in parallel with a spring (Fig.\ 2, inset) to determine $F^{act}(q)$. The contribution of each actuator to the generalized coordinates and spring models developed by Jafferis et al. \cite{16} are used for simplicity:

\[
F_s(V, q_{act}) = (f_1(\mathcal{G}) + f_2(\mathcal{G})q_{act})V \tag{3}
\]

where $F_s$ is the piezoelectric force, $K$ is the spring stiffness, $V$ is the AC drive voltage, and $q_{act}$ is the actuator tip deflection. The functions $f_1$, $f_2$, and $f_3$ depend only on constant geometric parameters, $\mathcal{G}$. The approximations in (3) have been experimental verified for the range of expected operating voltages (\sim 100-200 V) \cite{5}. The generalized external force can then be written as,

\[
F^{ext}(V, q) = F^b(q, \dot{q}) + B(q)^T (F_s(V, q) - K q). \tag{4}
\]

where $B^T$ is the Jacobian mapping actuator forces into generalized coordinates. Substituting (4) into (1) and (2) gives a complete set of differential-algebraic equations that capture the dynamics of the robot:

\[
\frac{d}{dt} D_2 \mathcal{L}(q, \dot{q}) - D_1 \mathcal{L}(q, \dot{q}) + C(q)^T \lambda = F^b(q, \dot{q}) + B(q)^T (F_s(V, q) - K q) \\
C(q) = 0. \tag{5}
\]

C. Contact Model

The tips of the robots four legs are modeled as point contacts. The contact forces are decomposed into directions normal and tangential to the running substrate. In the normal direction, collisions must obey a non-penetration constraint:

\[
\phi(q) \geq 0, \tag{6}
\]

where $\phi(q)$ is a function that returns the signed distance between the four leg-tips and the running substrate. Tangential forces are modeled using Coulomb friction, and they must satisfy the Maximum Dissipation friction Principle \cite{26}. This states that the friction force instantaneously maximizes the dissipation of kinetic energy, and is a generalization of the 2D concept of the friction opposing the direction of motion. Mathematically, this can be formulated as the following optimization problem:

\[
\begin{align*}
\text{minimize } & \quad \dot{q}^T D(q)^T b \\
\text{subject to } & \quad ||b|| \leq \mu \gamma,
\end{align*}
\]

where $b$ is the friction force, $\gamma$ is the normal force, $\mu$ is the coefficient of friction, and $D$ is the Jacobian mapping tangential contact forces into generalized coordinates. The norm constraint on $b$ ensures that the friction force lies within the Coulomb friction cone.

Instead of using the linear complementarity formulation described by Stewart and Trinkle \cite{33} to combine the normal and tangential contact constraints with the robot dynamics \cite{5}, we adapt the variational framework recently developed by Manchester and Kuindersma \cite{22}. This extends the linear complementarity formulation to higher-orders of integration accuracy, crucial for developing feasible locomotion plans for execution on a complex robot.

The variational framework requires approximating the Lagrange-D’Alembert principle with a quadrature rule before taking variational derivatives \cite{23}. The order of the resulting discrete equations-of-motion depends on the choice on quadrature rule, and we use the midpoint rule, which results in second-order accuracy. The normal contact constraints \cite{2} and the microrobot’s kinematic constraints \cite{2} are added to the discrete Lagrange-D’Alembert principle with the appropriate Lagrange multipliers, $\gamma$ (the normal force) and $\lambda$ (the constraint force), respectively. Taking variations with respect to the generalized coordinates $q_k$ then leads to the discrete Euler-Lagrange equations for our system:

\[
\begin{align*}
D_2 \mathcal{L}_d(h, q_{k-1}, q_k) + D_1 \mathcal{L}_d(h, q_k, q_{k+1}) + \\
\frac{1}{2} F^{ext}_d(h, q_{k-1}, q_k) + \frac{1}{2} F^{ext}_d(h, q_k, q_{k+1}) + \\
\frac{1}{2} (C(h, q_{k-1}, q_k) + C(h, q_k, q_{k+1}))^T \lambda_k + \\
N(q_{k+1})^T \gamma_k = 0.
\end{align*}
\]

Here $\mathcal{L}_d$ is the discrete Lagrangian for the robot, defined as the midpoint approximation of the integral of the continuous Lagrangian taken over a single time step. $F^{ext}_d$ is an analogous discretization of the generalized external force \cite{4}. Note that, in general, $\mathcal{L}_d$ and $F^{ext}_d$ depend on the choice of quadrature rule, and expressions for both are derived for a general system using the midpoint rule in \cite{22}.

\[
N(q)^T = (\partial \phi/\partial q)^T, \quad \text{and } C(q)^T = (\partial c/\partial q)^T \quad \text{are the Jacobians mapping normal contact forces and kinematic constraint forces into generalized coordinates, respectively. In addition to (5), solutions must satisfy the following constraints, known}
\]
as Karush-Kuhn-Tucker (KKT) conditions \[3\]:
\[
\begin{align*}
    c_d(q_k, q_{k+1}) &= 0 \\
    \gamma_k &\geq 0 \\
    \phi(q_{k+1}) &\geq 0 \\
    \gamma_k^T \phi(q_{k+1}) &= 0,
\end{align*}
\]
where \[9\] ensures that kinematic constraints are enforced. The three conditions in \[10\], collectively known as a complementarity constraint, prevent interpenetration and ensure that contact forces act only when bodies are in contact to push them apart. Such constraints are commonly denoted using the shorthand notation:
\[
0 \leq \gamma_k \perp \phi(q_{k+1}) \geq 0.
\]

The KKT optimality conditions for the Maximum Dissipation Principle \[7\] are approximated and discretized using the midpoint rule in a similar manner, resulting in three additional constraints: one equality constraint and two complementarity constraints,
\[
\begin{align*}
    g_1(h, q_k, q_{k+1}, \psi_k, \eta_k) &= 0 \\
    0 \leq \psi_k \perp g_2(\gamma_k, \beta_k) &\geq 0 \\
    0 \leq \beta_k \perp \eta_k \geq 0,
\end{align*}
\]
where \(\psi\) and \(\eta\) are Lagrange multipliers and the exact forms of \(g_1\) and \(g_2\) are derived in \[22\]. The tangential force and corresponding constraints are added to \[8\] and \[10\] to complete the dynamics model,
\[
\begin{align*}
    r(h, q_{k-1}, q_k, q_{k+1}, \gamma_k) + P(q_{k+1})^T \beta_k &= 0 \\
    g_1(h, q_k, q_{k+1}, \psi_k, \eta_k) &= 0 \\
    c(q_k, q_{k+1}) &= 0 \\
    0 \leq \gamma_k \perp \phi(q_{k+1}) \geq 0. \\
    0 \leq \psi_k \perp g_2(\gamma_k, \beta_k) &\geq 0 \\
    0 \leq \beta_k \perp \eta_k \geq 0.
\end{align*}
\]
Here \(r\) is the LHS of \[8\], and \(\beta\) and \(P\) are defined in \[22\] based on \(b\) and \(D\). Given \(q_{k-1}\) and \(q_k\), \[13\] can be solved to find \(\lambda_k, \beta_k, \psi_k, \eta_k, \) and \(q_{k+1}\).

D. Surface Characterization

The coefficients of static friction \(\mu\) between microrobot’s leg-tip and PTFE (Teflon), card-stock, and 1200 grit sandpaper are measured to complete the contact model. Experiments are conducted using a single leg (Fig. 3a) to closely replicate conditions during locomotion. Each surface is placed on an acrylic mounting plate and fastened to a six-axis force sensor (ATI Nano17 Titanium). The single leg is mounted on two micro-positioning stages and centered above the force sensor. Eight trials are run on each surface, and force data is recorded at 100 Hz using MATLAB’s XPC environment (MathWorks, MATLAB R2015a). Force traces for a representative trial are shown in Fig. 3b. The leg is manually lowered to pre-load force sensor to 35%-200% of the robot’s body weight (between 1 and 2 in Fig. 3b). The swing DOF is then actuated, generating a force in the \(x-y\) plane (3 in Fig. 3b), until the leg begins to slip (4 in Fig. 3b). Force data is filtered using an atausal low-pass Butterworth filter with a cutoff frequency of 10 Hz. The normal, \(F_n\), and static frictional, \(F_f\), forces are computed as:
\[
F_n = \Delta F_z
\]
\[
F_f = \sqrt{\Delta F_x^2 + \Delta F_y^2},
\]
where \(\Delta F_x, \Delta F_y, \) and \(\Delta F_z\) are the net forces between stages 1 and 4 in Fig. 3b in the \(x, y,\) and \(z\) directions, respectively. The friction force increases linearly with the normal force as anticipated (Fig. 3b). The mean and standard deviation for coefficients of friction averaged over the eight trials for Teflon, card-stock, and 1200 grit sandpaper are: 0.29 \(\pm\) 0.03, 0.51 \(\pm\) 0.07, and 1.02 \(\pm\) 0.20, respectively. Lines corresponding to these average friction coefficients are shown in Fig. 3.

III. Trajectory Optimization

The dynamics expressed in \[13\] are used as constraints in a direct trajectory optimization scheme. The trajectory optimization problem is posed as a standard nonlinear program (NLP) and solved using the constrained optimization solver SNOPT. The equality constraints from the three complementarity conditions in \[13\] are smoothed by replacing them with inequalities and introducing the slack variables \(s_k\) as in \[9\]. To encourage convergence of solutions towards satisfaction of the true complementarity constraints, we augment the cost function with a term that penalizes \(s_k\). The complete formulation of the trajectory optimization problem is stated.
in the following NLP:

\[
\begin{aligned}
\text{minimize } & \quad J(h, Q, V) + \alpha \sum_{k=1}^{N-1} s_k \\
\text{subject to } & \quad f(h, q_{k-1}, q_k, q_{k+1}, \lambda_k, \psi_k, \eta_k) = 0 \\
& \quad g(q_{k+1}, \lambda_k, \psi_k, \eta_k, s_k) \geq 0 \\
& \quad v_{\min} \leq v_k \leq v_{\max} \\
& \quad q_{\min} \leq q \leq q_{\max},
\end{aligned}
\]

where \( J \) is a cost function, \( \alpha \) is a positive scalar weighting parameter, and \( f \) and \( g \) are the equality and inequality constraints in the relaxed version of (13) found in [22]. Furthermore, \( Q \) is the set of all configuration knot points, \( q_k, V \) is the set of all control voltages, \( v_k \), and \( C \) is the set of all constraint-related variables, \( \gamma_k, \beta_k, \lambda_k, v_k, \psi_k, \eta_k \), and \( s_k \). The input voltages, \( v_k \), are bounded by \( v_{\min} = 0 \) V and \( v_{\max} = 225 \) V to increase actuator lifetime [16]. The flexure joint angles, a subset of \( q_k \), are bounded between \( q_{\min} = -\pi/4 \) and \( q_{\max} = \pi/4 \), which are conservative estimates of the maximum mechanical bend angles. The penalty on the slack variables in the cost function of (16), is an “exact penalty” that has theoretical convergence guarantees with finite values of \( \alpha \) [11]. In practice, we find good convergence with values of \( \alpha \) on the order of \( 10^2 \).

### A. Gait Optimization

We search for gaits near stride frequencies of 2 Hz, 10 Hz, and 30 Hz on three different surfaces: Teflon, card-stock, and 1200 grit sandpaper. The selected frequencies represent different operational regimes for the quadruped microrobot: quasi-static (2 Hz), near the \( z \)-natural frequency (10 Hz), and near the roll natural frequency (30 Hz) as discussed by Goldberg et al. [10]. Additionally, these nine gaits cover a wide-range of (ground) contact conditions (\( \mu \in [0.29, 1] \)), showcasing the versatility of our approach.

The NLP presented in (16) is modified to search for periodic state and input trajectories by enforcing periodicity constraints on all position and velocity decision variables except the \( x \)-position of the floating base. The algorithm minimizes the following cost function that encourages the robot to move approximately twice its stride length per cycle:

\[
J = (x_N - x_g)^T Q(x_N - x_g)^T + \sum_{i=2}^{N-1} \frac{c_1}{2} \Delta u_i^T \Delta v_i + \frac{1}{2} \Delta u_i^T \Delta u_i,
\]

where \( x_g \) is a goal state, \( Q \) is the identity matrix with \( Q_{11} \in [10, 50] \), \( c_1 \in [10, 50] \), and \( \Delta u_i = u_i - u_{i-1} \) and \( \Delta v_i = v_i - v_{i-1} \). The goal state is defined as \( x_g = [10, x_p]^T \), which corresponds to the robot translating forward slightly less than twice the maximum swing displacement (6 mm) of a single leg. The periodic subset \( x_p \) is set to the periodic subset of a state on a forward-simulated trot at the appropriate frequency, which is also used to initialize the optimization. The difference penalties are applied to discourage chatter in the state and inputs trajectory, which can alter behavior given the transmission’s high bandwidth (\( \sim 90 \) Hz). The standard quadratic penalty on inputs is not applied as it results in impractical gaits that barely lift the legs off the ground.

### B. Aperiodic Behaviors

We also used this variational trajectory optimization method to find state and input trajectories for a vertical jump. The following cost function, which encourages the robot to jump to a specific height, is minimized:

\[
J = (x_N - x_g)^T Q(x_N - x_g)^T + \sum_{i=1}^{N-1} \frac{1}{2} u_i^T R u_i.
\]

Here \( x_g \) is a goal state that specifies a final height of 24 mm (about one body height) with no body rotation or horizontal motion, and the quadratic input cost penalizes swing actuator voltages as fore/aft forces do not contribute significantly to a vertical jump.

### IV. LOCOMOTION EXPERIMENTS

The robot’s performance for trajectories found in Sec. III is evaluated in a controlled 20 cm \( \times \) 20 cm locomotion arena (Fig. 3). A proportional-derivative controller is implemented to track the desired positions of the robot’s four legs in the body-fixed frame, and an estimator is developed to provide low latency estimates of the leg positions (Fig. 4b).

#### A. Locomotion Arena

Input signals are generated at 2.5 kHz using a MATLAB xPC environment (MathWorks, MATLAB R2015a), and are supplied to the robot through a nine-wire tether. The xPC Target commands these signals through a digital-to-analog converter and custom high voltage amplifiers. Five motion capture cameras (Vicon T040) track the position and orientation of the robot at 500 Hz with a latency of 11 ms. A custom C++ script using the Vicon SDK enables tracking of the leg tips in the body-fixed frame as well. In addition, eight piezoelectric encoders (described below) provide low-latency estimates of actuator tip velocities at 2.5 kHz [17].

#### B. Piezoelectric Encoder Dynamics

Each piezoelectric encoder (Fig. 5) provides an estimate of a corresponding actuator’s tip velocity by computing the “mechanical” current produced in that actuator. This current, \( i_m \), is proportional to the actuator tip velocity \( \dot{x} \) and it can be computed for each actuator by applying Kirchoff’s law [19] to the measurement circuit in series with a standard lumped-parameter electrical model of an actuator [11]:

\[
i_m = \frac{V_{\text{sig}} - V_{\text{act}}}{R}\frac{\dot{x}}{C} - \frac{\dot{x}}{R}.
\]

The first term on the RHS is the total current drawn by a actuator, which is computed on the xPC target from measurements of the voltages before \( V_{\text{sig}} \) and after \( V_{\text{act}} \) a shunt resistor \( R_s \). The actuator is modeled as a capacitor, \( C \), resistor, \( R \), and current source, \( i_m \), in parallel. The voltage and
frequency dependent values of $R$ and $C$ have been computed for the range inputs by Jayaram et al. [17]. Furthermore, $\zeta$ is a blocking factor which accounts for imperfect measurements of $R$ and $C$, and is set to 1.57 as in [17].

C. Controller and Estimator Design

Since the motion capture measurement latency is a significant percentage of the gait period ($\sim$30% at 30 Hz), a filter is developed to provide low-latency leg position estimates in the body-fixed frame by using the “mechanical” current measurements from the piezoelectric encoders. These currents are scaled by an empirically determined proportionality constant, $X$, to estimate actuator velocities. For a particular leg, the SFB transmission kinematics are used to define a transformation, $H \in \mathbb{R}^{3 \times 2}$, from lift and swing actuator velocities to Cartesian leg velocities in the body-fixed frame. This map is used to estimate Cartesian leg velocity in the body-fixed frame, which are integrated over the duration of the latency to achieve low-latency leg position measurements.

$H$ (20) is indexed by achievable leg-$x$ and leg-$z$ positions (instead of lift and swing actuator positions) as they are directly estimated,

$$H(x, z) = \frac{\partial f}{\partial q_i} - \left[ \frac{\partial f}{\partial q_d} \right] \left[ \frac{\partial c}{\partial q_d} \right]^{-1} \frac{\partial c}{\partial q_i}.$$  (20)

Here $q_i$ are the independent generalized coordinates (actuator positions), $q_d$ are the dependent generalized coordinates (flexure joint angles), $c$ is the kinematic loop constraints, and $f$ is the kinematic mapping from lift and swing actuator positions to Cartesian leg position. We solve an inverse kinematics problem, posed as an NLP, to find values of $q_i$ and $q_d$ that result in desired leg-$x$ and leg-$z$ values. Twenty-one samples are used in each direction, and $H(x, z)$ is defined as a $6 \times 2$ look-up table stored on the xPC target.

This look-up table is then used in the following estimator:

$$\hat{q}[k]_{\text{leg}} = q[k - \epsilon]_{\text{leg}} + T_s \sum_{\kappa=k-\epsilon}^{k} \frac{1}{2}(\hat{q}[\kappa]_{\text{leg}} - \hat{q}[\kappa - 1]_{\text{leg}})$$

$$\hat{q}[k]_{\text{leg}} = H(x[k - 1], z[k - 1]) \begin{bmatrix} X^s & 0 \\ X^l & 0 \end{bmatrix} \begin{bmatrix} \dot{z}^s_{\text{m}}[k] \\ \dot{z}^l_{\text{m}}[k] \end{bmatrix}.$$  (21)

In the first equation, $\hat{q}[k]_{\text{leg}}$ is the estimated leg position, $q[k - \epsilon]_{\text{leg}}$ is leg position measured by the motion capture system with $\epsilon$ latency, $T_s$ is the sample rate of the xPC Target, and the summation is a trapezoidal integration of the estimated leg velocity over the duration of the latency. In the second equation, $H$ is indexed by the previous estimate of the leg-$x$ and leg-$z$ positions, $X^s$ and $X^l$ are the proportionality constants for the swing and lift actuators, respectively, and $\dot{z}^s_{\text{m}}[k]$ and $\dot{z}^l_{\text{m}}[k]$ are the mechanical currents for the lift and swing actuators, respectively.

These estimated leg positions are used in a simple controller as the swing and lift actuators independently control the leg-$x$ and leg-$z$ position at pre-resonant (<90 Hz) drive frequencies, respectively [9]. The feedback controller simply alters the feed-forward lift and swing input voltages ($V_0 = [V_0^s, V_0^l]$) based on the following control law:

$$V = \begin{bmatrix} K_s^s & 0 \\ 0 & K_f^l \end{bmatrix} V_0 + \begin{bmatrix} K_s^s & 0 \\ 0 & K_f^l \end{bmatrix} e + \begin{bmatrix} K_d^s & 0 \\ 0 & K_d^l \end{bmatrix} \frac{de}{dt}.$$  (22)

Here $V = [V_s, V_l]'$ are the input signals provided to the swing and lift actuators, and $e = [e_x, e_z]'$ is the error between the desired and estimated leg-$x$ and leg-$z$ positions. $K_p$, $K_d$, and $K_f$, are the proportional, derivative and feed-forward control gains, respectively, and, the superscripts $s$ and $l$ represent the lift and swing DOF, respectively.

D. Calibration

Eight amplification factors ($X$) from mechanical current (mA) to tip velocity in (in mm/s) are empirically computed for each actuator over the range of operating frequencies and voltages. $X$ is set to the value that minimizes the squared-error...
between the estimated leg velocities, and leg velocities computed by differentiating leg position measurements from the motion capture system and shifting them forward in time by the measured latency. The values of $X$ for each actuator is shown in Tab. I. Negative values indicated a $180\degree$ reversed orientation of the actuator with respect to the body-fixed frame.

### V. Results

We demonstrate a physical implementation of the generated trajectories for nine periodic gaits, and a dynamic jumping behavior (see attached video). In addition, we show that our simple joint-space controller tracks the desired leg trajectories in the body-fixed frame for a range of frequencies on a wide variety of surfaces.

#### A. Gait Optimization

Each gait is executed for 15 cycles with an initial voltage ramp, and the control gains are manually tuned. The feed-forward gains remain close to unity (ranging form 0.9 – 1.1), reflecting utility of the planned inputs. The mean per-cycle forward velocity for the closed-loop experimental trials (blue) is compared with the optimized trajectory (orange) and a heuristically tuned gait with same frequency and average input power (gray) in Fig. 6b.

The 2 Hz closed-loop experimental trajectories achieve an average velocity of $9.77\, \text{mm/cycle}$, which is within 5% of the goal speed of $10\, \text{mm/cycle}$. These gaits also perform 26% better than the heuristically tuned gaits, which achieve an average velocity of $7.71\, \text{mm/cycle}$. In addition, the planned body pose closely matches that which is executed by the robot (card-stock trial shown in Fig. 6a), demonstrating that the model captures most of the robot’s salient dynamic properties. Finally, the closed-loop leg trajectories (front left leg depicted in Fig. 6c) closely match the optimized trajectories in air ($z > 0$). However, the robot is unable to push as forcefully into the ground as planned, most likely because of unmodeled serial compliance in the transmissions. This behavior is also consistent for all 10 and 30 Hz trajectories.

At 10 Hz, the closed-loop experimental trajectories achieve an average velocity of $8.98\, \text{mm/cycle}$, which is close to the desired velocity and 10% faster than the heuristically tuned gaits. The card-stock gait at this frequency achieves the fastest per-cycle velocity recorded for this quadrupedal microrobot at $10.87\, \text{mm/cycle}$; however, the other two gaits perform slightly worse than expected. This is most likely due to discrepancies between the planned and executed floating base trajectories (see attached video). Finally, the average velocity for the 30 Hz closed-loop gaits is slower at $4.24\, \text{mm/cycle}$. The closed-loop experiments on sandpaper and card-stock are still 20% percent faster than the heuristically tuned gaits and within 20% of the predicted optimized velocities; however, performance is poor on Teflon. This frequency (near the roll resonance) is particularly challenging for locomotion using the laterally asymmetric trot gait, and the optimizer has difficulty (especially on Teflon) finding gaits that move $10\, \text{mm}$ per cycle.

#### B. Aperiodic Behaviors

We also use this method to execute four vertical jumps on a card-stock surface. The average height achieved is $22\, \text{mm}$, which is within 10% of the goal jump height of $24\, \text{mm}$ (Fig. 7). The maximum height achieved is $22.65\, \text{mm}$, corresponding to a jump height of $9.96\, \text{mm}$ (78% body height). Manufacturing imperfections lead to lateral asymmetries that cause the robot to roll during the jump, but it does not drift.
much from the initial position, with initial \(x\), \(y\) and \(yaw\) values of \(-3.25 \pm 0.57\) mm, of \(-0.15 \pm 1.98\) mm, and of \(11.12 \pm 8.56\) deg, respectively.

VI. DISCUSSION

A. Performance Improvements

This model-based approach yields improvements over previous experimental results collected in [5][10]. Specifically, the average velocity of \(7.55 \pm 2.79\) mm/cycle achieved across all nine gaits is comparable to carefully tuned previously measured experimental velocity of \(8.2\) mm/cycle (on cardstock). Even the three slower \(30\) Hz gaits move on average \(30\%\) faster than previously recorded trots at similar frequencies on a card-stock surface. This is notable considering the range of operating frequencies and variety in running substrates. Additionally, the robot is able to achieve a new highest velocity of \(10.87\) mm/cycle, and demonstrate the first controlled vertical jump of \(9.6\) mm (78\% of body height).

B. Quality of Optimized Trajectories

We evaluated the quality of the periodic trajectories by measuring the normalized average slip, \(\bar{s}\) defined as:

\[
\bar{s} = \frac{1}{4} \sum_{i=1}^{4} \frac{\int_{0}^{T} |R(t)\hat{q}_{ix}(t)|dt}{\int_{0}^{T} |R(t)\hat{q}_{ix}(t)|dt}.
\]

Here \(\hat{q}_{ix}\) is the estimated \(x\)-velocity of the \(i\)th leg in the body-fixed frame, \(R\) is the rotation matrix from body to world coordinates, \(T\) is the gait period, and \(\xi\) is the set of times for which \(R_{ij}^{\xi} < 0\). Intuitively, normalized slip for a single leg is the total distance traveled backwards normalized by the total distance traveled in the world frame, and we present an average value for all four legs. Higher values of \(\bar{s}\) indicate increased backwards motion of the legs, decreased forward propulsion, and consequently, reduced performance. The average value of \(\bar{s}\) is \(0.08 \pm 0.05\) (\(n = 9\)) for the optimized trajectories. This small magnitude of slip is expected since we demand high performance from the robot with per-cycle stride lengths approaching twice its maximum swing displacement. The closed-loop experimental trajectories slipped slightly more, with an average \(\bar{s}\) of \(0.20 \pm 0.05\) (\(n = 9\)), and is one of the factors that could have resulted in decreased performance. The optimizer also finds an intuitive jumping trajectory where all four legs first build potential energy, and then simultaneously push into the ground (see attached video).

C. Limitations

Due to the nonconvexity of the trajectory optimization problem, it is possible for the optimizer to get stuck in poor local optima depending on the initial values of the decision variables. This is most clear for the \(30\) Hz frequencies, where other (laterally symmetric) gaits have been shown to achieve higher per cycle velocities [10]. Furthermore, our current MATLAB implementation requires several minutes to compute the plans described in the previous sections, and significant speed improvements could be made with a C++ implementation that exploits the sparsity of the problem.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we develop and evaluate a framework for modeling, planning, and controlling dynamic behaviors for legged microrobots. We developed a full three-dimensional dynamic model of a complex quadrupedal microrobot with eight control inputs, a 76 dimensional state, and 24 kinematic position constraints. We also adapt a state-of-the-art variational contact-implicit trajectory optimization algorithm to generate physically accurate whole-body locomotion plans for a variety of operating conditions. These locomotion plans are executed on the robot and result in improve performance, including the fastest recorded per-cycle velocity for this robot and a demonstration of the first controlled vertical jump.

Future work can be divided into two major categories: improvements to the existing variational framework, and the development of increasingly dynamic and novel behaviors for legged microrobots. As other authors have observed [25], the contact mode trajectory often changes over longer timescales than the state and input trajectories. This is particularly true for microrobots with high frequency passive dynamics. Explicitly encoding this into the NLP formulation by using 3rd and 4th-order time-stepping methods could aid convergence and avoid unnecessary contact transitions without sacrificing richness in the contact force trajectories. This improved framework be paired with a whole-body locomotion controller to plan increasingly dynamic behaviors, including climbing and resonant running [6], without careful initialization.

REFERENCES


