

# Evaluation of Decentralized Reactive Swing-Leg Control on a Powered Robotic Leg

Alexander Schepelmann<sup>1</sup>, Jessica Austin<sup>1</sup>, and Hartmut Geyer<sup>1</sup>

**Abstract**—Animals and robots balance dynamically by placing their feet into proper ground targets. While foot placement controls exist for both fully robotic systems and powered prostheses, none enable the dynamism and reactivity of able-bodied humans. A control approach was recently developed for an ideal double pendulum dynamical system that places feet into ground targets for a wide range of initial conditions and in the presence of significant locomotion disturbances. While its performance in simulation make it an attractive candidate to control legged robotic systems, it is unclear if the approach can be used on real-world systems. In this paper we transfer and evaluate the approach on robotic hardware. Our results show that the controller can be transferred to robotic systems and allow them to achieve comparable foot placement accuracies to an ideal double pendulum simulation, both when motion is undisturbed, as well as when obstacles are encountered in early, mid, and late swing. These results suggest that the proposed approach is a potential alternative control method for legged robots and powered prostheses, enabling recovery from sudden swing-leg disturbances such as trips and unexpected obstacle encounters. The results also point out the need for additional considerations when tuning the controller in order to generate human-like swing trajectories and durations.

## I. INTRODUCTION

Animals and legged robots balance dynamically by placing their feet into proper ground targets [1][2]. Ground targets that stabilize locomotion in the presence of disturbances can be identified with simple locomotion models, like the linear inverted pendulum model for walking [3][4] and the spring-mass model for running [5][6]. Robotic systems then place their feet into identified ground targets in several ways.

Fully robotic systems pre-plan and execute full-body trajectories using inverse dynamics and kinematics [7][8]. While these approaches enable robots to climb stairs [7] and react to push disturbances [9][10], they require estimates of the robot’s full state and an accurate system model. Such approaches can therefore not be used to control robotic assistive devices where the human user’s state is unknown.

Human-in-the-loop locomotion controls primarily replay motion patterns extracted from healthy human gait, including joint impedance [11][12], joint motion [13][14], or a combination of the two [15]. These patterns implicitly encode foot placements for steady-state level- and sloped-ground walking at various speeds. Some research has explored extensions to impedance-based controls to identify stumbles and explicitly

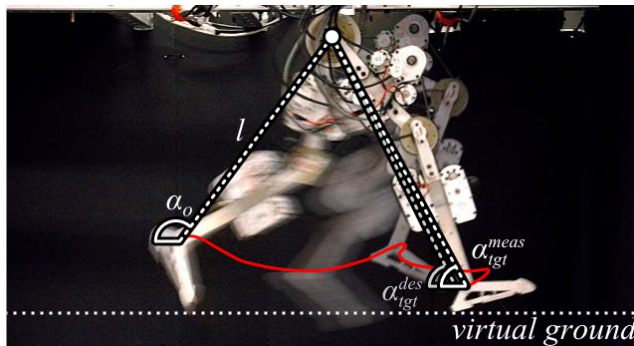


Fig. 1. Swing-leg control experiment. The controller issues joint torque commands to regulate the length  $l$  of a single segment, virtual leg between the robot’s hip and ankle, moving it from an initial position  $\alpha_0$  to a target position  $\alpha_{tgt}^{des}$  when making contact with a virtual ground (dotted). Solid: Trajectory traced out by ankle point during experiment.

regulate foot placements during such disturbances [16][17], but no conclusive results have been demonstrated.

Heuristic controls that regulate foot placements based on simple locomotion models are alternative control approaches for fully robotic systems and powered prostheses. Commands are not dependent on a system’s full state, but other metrics like system energy [18], leg angle of attack at touchdown [6], and states of virtual muscle actuators [21]. Such controllers allow simple robots to traverse rough terrain equivalent to their leg length [2][19] and have been used in an active ankle-foot prosthesis to regulate joint torque during stance [20].

We recently proposed a swing-leg controller based on double pendulum dynamics that *explicitly* regulates foot placement into desired ground targets, formulated as a desired leg angle of attack at touchdown [22]. When applied to an ideal frictionless pendulum simulation, this controller places feet into desired target angles for a wide range of initial conditions and in the presence of significant locomotion disturbances, even with translational hip accelerations seen in human locomotion. The controller was later re-formulated to use virtual muscle actuators instead of ideal torque sources [23], and incorporated into a planar, muscle-reflex based locomotion model [24]. Despite lacking central processing, this model walked steadily over level and rough terrain, and up stairs without parameter changes. The controller’s performance in simulation make it an attractive candidate to control both fully robotic systems and powered prostheses, but it is unknown if this approached can be used in hardware.

In this paper, we present work to transfer this swing-leg control approach to robotic hardware (Fig. 1) and quantify its ability to execute foot placements into desired ground

\*This work is supported by the Eunice Kennedy Shriver National Institute of Child Health & Human Development under award no. 1R01HD075492.

<sup>1</sup>A. Schepelmann, J. Austin, and H. Geyer are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213, USA. {aschepelmann, jaustin, hgeyer}@cmu.edu

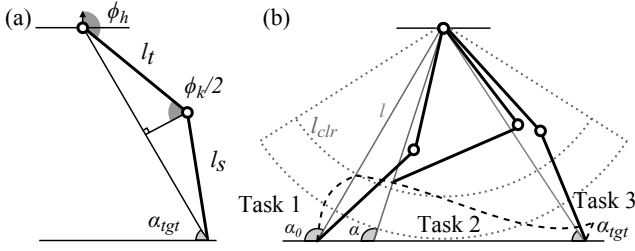


Fig. 2. Swing-leg control. (a) Model geometry. (b) Task sequence.

targets, both when the leg's motion is undisturbed and when unknown obstacles are in the leg's path. A summary of the swing-leg controller is presented in Section II. The hardware platform Robotic Neuromuscular Leg 2 (RNL2), a dynamically scaled, antagonistically actuated robotic leg is introduced in Section III. A simulation of RNL2 hardware is used to transfer the swing-leg controller to our robot, which is discussed in Section IV. Simulation and hardware experiments to evaluate controller performance are presented in Section V. Hardware experiments show that the controller can be transferred to robotic systems and achieve foot placements into desired ground targets with comparable accuracy to an ideal double pendulum simulation, both when the robot's motion is undisturbed, as well as when it encounters obstacles in early, mid, and late swing. Results and future work are discussed in Section VI.

## II. SWING-LEG CONTROLLER

The swing-leg controller we proposed in [22] is a high level controller based on ideal double pendulum dynamics to achieve foot placements into specified ground targets  $\alpha_{tgt}$  via combined hip and knee control. Hip and knee angles  $\phi_h$  and  $\phi_k$  are reinterpreted as a length  $l$  and angle  $\alpha$  of a single-segment virtual leg (Fig. 2a). Assuming equal thigh and shank lengths  $l_t = l_s$ ,  $\alpha = \phi_h - \frac{\phi_k}{2}$  and  $l = 2l_t \sin(\frac{\phi_k}{2})$ .

Three sequential tasks must be accomplished to place the foot during swing: leg flexion to achieve ground clearance, leg advancement to a placement target, and leg extension until ground contact (Fig. 2b). While this sequence could be realized by tracking predefined trajectories, heuristic controls are used for three reasons. First, tracking would require extensive predefined motion libraries that may not handle disturbances. Second, heuristic controllers can exploit passive dynamics to lower joint torques required to realize swing-leg motions. Third, joint controls can be largely decoupled to modularize the controller for use in multiple prosthesis configurations. For this reason, the controller is structured as two functionally distinct hip and knee joint controllers.

### A. Hip Control

Hip control is active throughout swing. Its primary function is to move the leg angle into  $\alpha_{tgt}$ , given by

$$\tau_h^\alpha = k_p^\alpha (\alpha_{tgt} - \alpha) - k_d^\alpha \dot{\alpha} \quad (1)$$

where  $k_p^\alpha$  and  $k_d^\alpha$  are proportional and derivative gains. During leg extension in late swing, the hip control receives an additional input  $\tau_h^{iii}$ , detailed in the next section.

### B. Knee Control

The knee control's primary function is to regulate  $l$ . Its control is separated into three tasks, shown in Fig. 2b, and can be realized as a finite state machine.

Passive dynamics are exploited to accomplish task one, leg flexion to achieve ground clearance during swing. Hip control (1) initially generates negative hip accelerations to drive the leg toward  $\alpha_{tgt}$ , resulting in passive knee flexion if  $\dot{\alpha} > 0$ . If  $\dot{\alpha} \leq 0$ , there is no passive knee motion; in such cases, active knee flexion, proportional to how fast the leg moves forward, is applied. The resulting control of task one is

$$\tau_k^i = \begin{cases} k^i \dot{\alpha} & \dot{\alpha} \leq 0 \\ 0 & \dot{\alpha} > 0 \end{cases} \quad (2)$$

where  $k^i$  is a proportional gain.

Task two, which holds the knee while the hip controller moves the leg toward  $\alpha_{tgt}$ , triggers once  $l$  becomes less than a predefined clearance length  $l_{clr}$ . Knee flexion ( $\dot{\phi}_k \leq 0$ ) is damped using a pure damping input. Knee extension uses a modulated damping input which allows passive extension to occur when  $\alpha$  approaches  $\alpha_{tgt}$ , but prevents premature landing if the knee extends faster than the overall leg angle. The resulting control of task two is

$$\tau_k^{ii} = \begin{cases} -k^{ii} \dot{\phi}_k & \dot{\phi}_k \leq 0 \\ -k^{ii} \dot{\phi}_k (\alpha - \alpha_{tgt}) (\dot{\phi}_k + \dot{\alpha}) & \dot{\phi}_k > 0 \ \& \ \dot{\phi}_k > -\dot{\alpha} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where  $k^{ii}$  is a proportional gain.

Control task three is made up of two components. The first component stops swing and extends the leg into  $\alpha_{tgt}$  when  $\alpha$  passes the threshold  $\alpha_{thr} = \alpha_{tgt} + \Delta\alpha_{thr}$ . This task, inspired by nonlinear contact models [21][25], generates a stopping knee-flexion torque

$$\tau_k^{iii} = \begin{cases} -k^{stp} (\alpha_{thr} - \alpha) (1 - \frac{\dot{\alpha}}{\dot{\alpha}_{max}}) & \alpha < \alpha_{thr}, \dot{\alpha} < \dot{\alpha}_{max} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where  $\dot{\alpha}_{max}$  is a parameter describing the leg's maximum return velocity at which reaction forces are developed. To cancel this torque's effect on the hip motion, a hip torque  $\tau_h^{iii} = -\tau_k^{iii}$  is applied during this control task. The second component activates once the leg has slowed to zero angular velocity,  $\dot{\alpha} = 0$ , and adds an active knee extension torque to ensure that the leg makes ground contact at the end of swing

$$\tau_k^{iii'} = \tau_k^{iii} + k^{ext} (l_0 - l), \quad (5)$$

where  $k^{ext}$  is a proportional gain and  $l_0 = l_s + l_t$ .

## III. ROBOTIC NEUROMUSCULAR LEG 2

RNL2 is a dynamically scaled, antagonistically actuated robotic leg with joint compliance (Fig. 3a). Its weight, size, and actuation requirements are based on dynamically scaled segment masses, lengths, and capabilities of virtual muscles in a planar, muscle-reflex based walking model [21]. The dynamic scaling procedure used to define the robot's

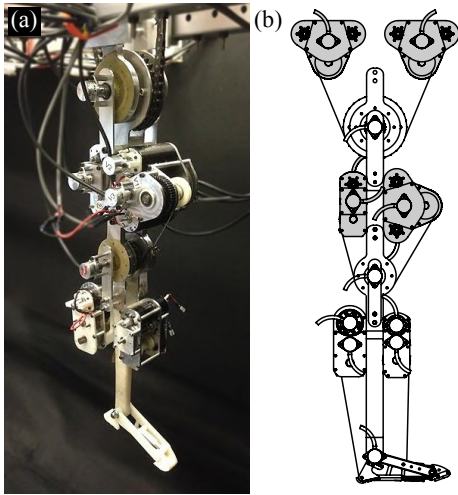


Fig. 3. The Robotic Neuromuscular Leg 2 (RNL2). (a) Hardware implementation. (b) Actuation schematic. Active swing-leg SEAs highlighted in gray. CW from top: Hip extensor, hip flexor, knee extensor, knee flexor.

mechanical and actuation targets is described in [26]. RNL2's thigh weighs 4.37kg and has a length of 23cm. Its shank weighs 1.88kg and has a length of 27cm. The following sections describe RNL2's electromechanical design.

#### A. Actuation and sensing

Four of RNL2's cable-driven series elastic actuators (SEAs) [27] are active during swing-leg control, acting on the robot's hip and knee joints as mono-articular antagonistic pairs (Fig. 3b). Using a cable-drive instead of a direct-drive to actuate the joints provides several design advantages. First, SEAs can be located away from the robot's joints, ideal for realizing more human-like segment mass distributions. Second, cable-drives can go slack, which allows them to act across compliant segments and enables truly passive dynamics. Third, they enable antagonistic actuation, which, when combined with nonlinear stiffening springs, can be used to modulate joint stiffness during actuator co-contraction [28].

The maximum joint torque  $\tau_j^{max}$ , maximum joint speed  $\dot{\theta}_j^{max}$ , total gear ratio  $n_{tot}$ , and spring stiffness  $k_s$  of each active SEA is summarized in Tab. I. With a later technology transfer to prosthetic and orthotic devices in mind, RNL2 uses electric DC motors in its SEAs, selected to meet desired maximum joint torque and speed targets while remaining within the specified size and weight envelope. The hip extensor, hip flexor, and knee extensor use two mechanically coupled DC motors for compactness (RE40: Maxon Motor AG); the knee flexor uses a single DC motor (RE30: Maxon Motor AG). All actuators follow the same drivetrain layout, with torsional springs that serve as the SEAs' compliant elements located after the first gear stage (Fig. 4a) Off-the-shelf spring couplers (Ruland Manufacturing Co.: Hip extensor, hip flexor, and knee extensor: MWC25-6-6-SS. Knee flexor: FCMR19-5-5-A.) are used as the SEAs' compliant elements.

Sensing and actuator control is implemented as a real-time, 1kHz system using xPC Target software (MathWorks, Inc.) and EtherCAT motor controllers (DZEANTU-020B080B:

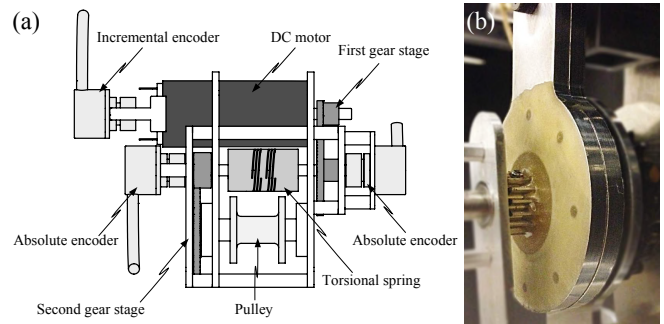


Fig. 4. SEA drivetrain schematic and joint compliance. (a) Knee flexor SEA shown for illustration purposes. A torsional spring used to measure SEA torque is located between the first and second gear stages. A cable connects the SEA pulley to the robot's joint, creating the third gear stage. (b) Hardware implementation of translationally compliant joint.

TABLE I  
RNL2 SEA CAPABILITIES

	Hip extensor	Hip flexor	Knee extensor	Knee flexor
$\tau_j^{max}$ (Nm)	90	90	45	10
$\dot{\theta}_j^{max}$ (rpm)	109	109	217	200
$n_{tot}$	72	72	36	42.5
$k_s$ (Nm/rad)	44.5	48.86	49.28	23.86

Advanced Motion Controls). SEA torque measurements are realized with two absolute rotary encoders (RM22SC 13B: Renishaw PLC) located on either side of the spring and are fed asynchronously to the target machine using a microcontroller (ATmega328-PU: Atmel Corporation). An additional incremental encoder on each DC motor shaft measures motor velocity (RM22I 09B: Renishaw PLC).

#### B. Joint compliance

Humans are not rigidly coupled kinematic chains, possessing interjoint cartilage and soft tissue around bones. To capture this aspect in RNL2, the robot incorporates a translationally compliant joint design at its hip and knee (Fig. 4b). Encoders are located on each joint shaft, measuring hip and knee position during leg movement.

## IV. RNL2 SIMULATION

A simulation of RNL2 is developed in Simulink SimMechanics (Mathworks, Inc.) (Fig. 5a), which models the robot at the individual component level. It is used to transfer the proposed heuristic swing-leg control approach to hardware, serving as a tool to evaluate if it can be applied to robotic systems, and how specific control components need to be implemented to account for hardware constraints. The simulation's software and control architecture is shown in Fig. 6. The implementation of each control level is outlined in the following sections. To capture behavior imposed by the proposed swing-leg controller's discrete-time execution when controlling robotic hardware, simulated high- and mid-level control loops are constrained to run at 1kHz.

#### A. High-level control

The high-level control block contains the swing-leg controller described in Section II and generates desired net

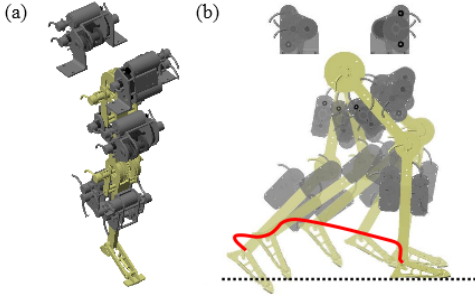


Fig. 5. RNL2 simulation. (a) Screenshot of RNL2 model. (b) Undisturbed trajectory experiment motion,  $\alpha_{gt} = 70\text{deg}$ . Solid: Traced ankle point trajectory. Dotted: Virtual ground.

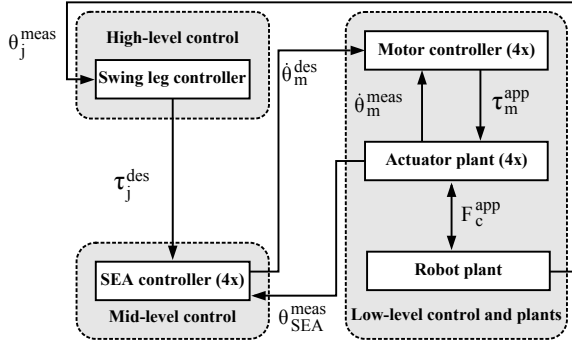


Fig. 6. RNL2 software and control architecture.  $\theta_j^{\text{meas}}$ : measured joint angles.  $\tau_j^{\text{des}}$ : desired joint torques.  $\theta_m^{\text{des}}$ : desired motor velocities.  $\theta_m^{\text{meas}}$ : measured motor velocities.  $\tau_m^{\text{app}}$ : applied motor torques.  $F_c^{\text{app}}$ : force transmitted by cable drive.  $\theta_{SEA}^{\text{meas}}$ : measured spring angles.

joint torques  $\tau_j^{\text{des}}$  based on  $\alpha$  and  $\dot{\alpha}$  calculated from joint measurements of the robot's hip and knee. Since RNL2's cable-driven SEAs can only pull on a joint, generated high-level torques are distributed to the extensor and flexor actuators based on sign, with positive torques representing extension torques. To compensate for instantaneous torque changes commanded by the swing-leg controller, which could result in system instability, commanded torques are low-pass filtered ( $f_c=25\text{Hz}$ ).

### B. Mid-level control

Desired actuator torques are passed to the mid-level control block that regulates SEA torques. This block generates desired motor velocity signals using the velocity-based controller detailed in [29]. Desired motor velocities are sent to the low-level control block containing motor controller simulations and hardware models.

### C. Low-level control & hardware plant

The low-level control block contains a simulation of RNL2's motor controllers, as well as plant models of physical hardware. Due to the motor controllers' 20kHz operating frequency, their dynamics, as well as the rest of the hardware, are modeled as continuous time systems.

Motor electrical dynamics are modeled as

$$V - iR - L \frac{di}{dt} - k_{EMF} \dot{\theta}_m = 0 \quad (6)$$

TABLE II  
RNL2 DRIVETRAIN AND JOINT FRICTION COEFFICIENTS

	$\mu_1$ (Nm)	$\mu_2$ (Nm)	$\mu_3$ (Nm)	$v_j$ (Nms/rad)
Hip extensor	0.0024	0.0026	0.0077	0.1547
Hip flexor	0.0035	0.0010	0.0029	0.1547
Knee extensor	0.0004	0.0004	0.0011	0.3900
Knee flexor	0.0019	0.0002	0.0006	0.3900

where  $V$ ,  $i$ ,  $R$ ,  $L$ ,  $k_{EMF}$ , and  $\dot{\theta}_m$  are the motor voltage, current, resistance, inductance, back-EMF constant, and motor velocity [30]. A  $\pm 48\text{V}$  saturation is applied to the model to simulate voltage limits of RNL2's motor controllers. Voltage commands are generated with a PID loop operating on the difference between desired and measured motor velocity. Dissipative no-load currents are subtracted for each motor before calculating transmitted torque.

Drivetrain stages are modeled in SimMechanics as separate physical, interacting bodies. Rotational inertias of each drivetrain stage are estimated from CAD models (SolidWorks 2012: DSS Corp.). Coulomb friction  $\tau_f^c$  is applied to each bearing stage, whose coefficients are calculated through linear regression of experimental characterization data using the equation  $\tau_f^c = \mu_i \text{sgn}(\theta_m)$  where  $\mu_i$  is the offset of Coulomb friction at each drivetrain stage. Viscous friction is applied to the robot's hip and knee joints; joint viscous friction coefficients  $v_j$  are calculated through linear regression of passive pendulum motion experiments. Identified friction coefficients used in the model are shown in Tab. II. The torsional stiffness of each SEA spring is calculated from testbed experiments using analog compression load cells (FC22, 100lbf: Measurement Specialties); identified spring constants are shown in Tab. I. RNL2's cable drives, which connect the robot's SEA drivetrain outputs to its joints, are modeled as series spring dampers with a stiffness  $k_c = 10,000\text{N/m}$  and damping  $v_c = 500\text{Ns/m}$ .

### D. Tuning

All simulation gains are tuned using CMA-ES [31], with a cost function to minimize the sum of squared differences between desired and measured signals at each control level. Gains are tuned hierarchically starting at the lowest level, ensuring that the motor controllers can follow commanded velocity signals. Mid-level SEA gains are then tuned to ensure that the actuators can realize commanded torques. Finally, high-level gains are optimized to minimize the difference between the desired and measured  $\alpha_{gt}$ . As in [22], all gains are tuned for undisturbed motion with  $\alpha_{gt} = 70\text{deg}$ . These gains are used for all experiments.

Hardware gains are also tuned hierarchically. Hardware gains are manually tuned. High-level control gain tuning uses the optimized high-level simulation gains as a starting point.

## V. EXPERIMENTS

A simulation of the swing-leg controller applied to an ideal double pendulum with human-sized segment mass, length, and inertia [22] is used as a baseline to evaluate behavior generated by the swing-leg controller running on RNL2 in

TABLE III

MEAN PLACEMENT ERROR FOR  $\alpha_{gt}$  RANGE: 65DEG TO 85DEG

Disturbance	None	Early	Mid	Late
Ideal Sim.	-1.47±0.61	-3.79±0.30	-4.27±0.55	-5.90±0.85
RNL2 Sim.	1.17±0.68	1.24±1.61	1.40±1.39	1.26±1.08
RNL2 Hrdw.	1.17±3.69	-2.70±4.33	-1.49±3.73	-3.51±3.37

TABLE IV

MEAN SWING TIME IN MS FOR  $\alpha_{gt}$  RANGE: 65DEG TO 85DEG

Disturbance	None	Early	Mid	Late
Ideal Sim.	394±5	529±3	593±6	516±86
RNL2 Sim.	328±9	366±14	356±25	334±13
RNL2 Hrdw.	481±102	640±103	583±129	546±70

simulation and hardware. In these experiments, RNL2 is suspended from its hip in a rigid mounting cage. While this setup eliminates dynamic effects due to trunk motion, previous simulation work showed that trunk dynamics do not negatively impact controller performance [24]. For control purposes, RNL2 is assumed to have equal, nominal shank and thigh lengths  $l_t = l_s = 27\text{cm}$  and  $l_{clr} = 4\text{cm}$ .

During experiments, the robot starts at a neutral position (Fig. 3). A feed-forward torque that is constant across all experiments moves the leg into an initial pose, which also imparts initial joint velocities. After 250ms, swing-leg control initializes, measuring  $l$ ,  $\alpha_0$ , and the virtual ground location defined by the robot's initial pose (Fig. 1). The controller then executes. Motion continues until the foot point again makes contact with the virtual ground location.

Two sets of experiments are used to evaluate the controller in simulation and hardware. Undisturbed motion experiments (Fig. 1) test the controller's ability to place feet into desired ground targets for unimpeded swing. Disturbed motion experiments simulate tripping and test the controller's ability to place feet into desired ground targets when the robot encounters an unexpected obstacle in early, mid, and late swing (Fig. 7). In hardware experiments, the obstacle is a 500g block on a set of rockers, approximating an impulse disturbance as the robot knocks the obstacle over after collision during swing. In simulation, obstacle collision is modeled using a horizontal force generated by a nonlinear contact model [21], whose parameters are estimated from the material properties of the leg and obstacle. For disturbed motion experiments the robot's foot is removed so that disturbances occur at the ankle, better emulating the disturbance condition experienced by the ideal double pendulum simulation. The foot's contribution to total shank mass and inertia are considered negligible. Trip obstacles are positioned as shown in Fig. 8. Experiments are conducted at 5deg  $\alpha_{gt}$  increments for undisturbed and all obstacle placement conditions, with 5 hardware trials for each condition<sup>1</sup>. Data from each obstacle position-desired target angle pair are grouped as early, mid, and late motion disturbances depending on whether they occur in the first, middle, or last third of swing.

<sup>1</sup>Due to hardware failure,  $n = 1$  for  $\alpha_{gt} = 85\text{deg}$  early swing disturbance and  $n = 4$  for  $\alpha_{gt} = 80\text{deg}$  mid swing disturbance conditions.

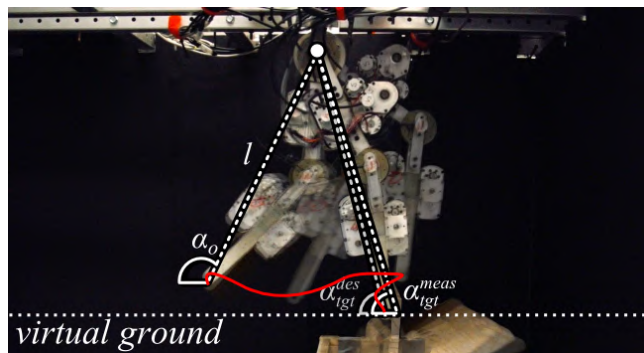


Fig. 7. Example disturbed swing-leg control experiment. Shown:  $\alpha_{gt} = 70\text{deg}$ , late disturbance.

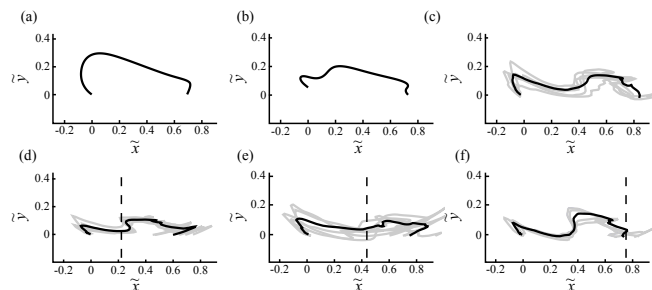


Fig. 8. Foot point trajectories of  $\alpha_{gt} = 70\text{deg}$  experiments normalized by respective total leg length  $(\tilde{x}, \tilde{y})$ . Disturbance type noted in parentheses. Black: Mean trajectories. Gray: Individual trials. Dashed: Obstacle location. (a) Ideal double pendulum (none) (b) RNL2 sim. (none) (c) RNL2 hrdw. (none) (d) RNL2 hrdw. (early) (e) RNL2 hrdw. (mid) (f) RNL2 hrdw. (late)

Mean simulation and hardware foot placement errors and swing times for ground target experiments between  $\alpha_{gt} = 65\text{deg}$  to  $85\text{deg}$ , corresponding to various step lengths during walking, are shown in Tab. III and Tab. IV, respectively.

## VI. DISCUSSION & FUTURE WORK

In both simulation and hardware, RNL2 places feet with comparable accuracy to the ideal double pendulum for all tested conditions (Tab. III), suggesting that the controller can accurately regulate foot placement of robotic legs. Hardware mean placement error either improves or is within the standard deviation of the ideal double pendulum simulation for all tested conditions.

Foot point trajectories of the double pendulum simulation, RNL2 simulation, and hardware experiments for  $\alpha_{gt} = 70\text{deg}$  experiments are shown in Fig. 8. Disturbed hardware trajectories suggest that the controller makes the robot execute a human-like foot elevation strategy when encountering obstacles in early swing, indicated by retraction of the foot point after it collides with the obstacle [32]. Whether the controller also causes execution of a lowering strategy for late swing obstacles is unclear and warrants further investigation. Though the step length for late obstacle encounters is shorter than the undisturbed case, characteristic of lowering strategy behavior, the step length for other disturbed swing motions are shorter than the undisturbed case as well.

While the magnitude of normalized step length is the same for all experiments, foot point height during swing, especially

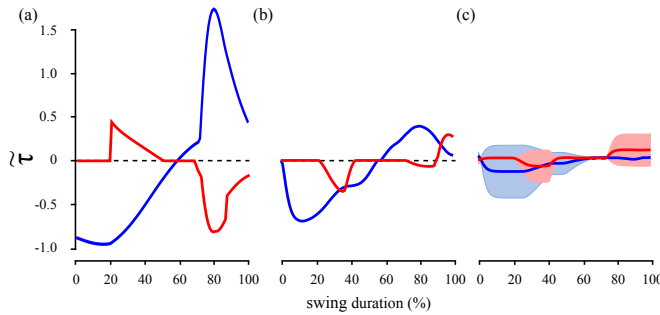


Fig. 9. Commanded torques for undisturbed  $\alpha_{gt} = 70\text{deg}$  experiments normalized by product of respective total leg mass, length, and gravity ( $\bar{\tau}$ ). Blue: Hip. Red: Knee. (a) Ideal double pendulum. (b) RNL2 sim. (c) RNL2 hrdw. Light blue: S.D. of mean hip torque. Pink: S.D. of mean knee torque.

early swing, is less pronounced in both the RNL2 simulation and hardware compared to the ideal double pendulum simulation. Comparing commanded joint torques between these systems (Fig. 9) reveals that both the characteristic shape and relative magnitude of the commanded torques are different between the ideal double pendulum and RNL2 systems, which plausibly results in the less pronounced ground clearance. This behavior is likely the result of the cost function used to tune the RNL2 simulation, whose gains served as a tuning starting point of the hardware gains. Whereas the swing-leg controller gains for the ideal double pendulum simulation were hand-tuned [22], the cost function used to tune the RNL2 gains did not include an explicit term to consider the overall cosmesis of the motion.

Hardware swing duration also exceeds dynamically scaled goals. Based on RNL2's dynamic scaling, swing should be approximately 30% faster than the ideal double pendulum's human-sized motion. This discrepancy can again be attributed to the lack of explicit consideration for factors besides overall placement accuracy when tuning the gains. These results suggest the need for additional cost terms when tuning robot control gains that trade off between placement accuracy, human-like motion, and swing execution time.

We immediately plan to transfer the swing-leg controller's neuromuscular formulation [23] to RNL2 to investigate benefits of multi-articulation for robotic and prosthetic systems. RNL2 is currently being modified to include a foot segment and multi-articular SEAs to match the actuator layout in [21]. To capture mechanical characteristics of muscle, we have developed a synthesis method for compact, nonlinear springs with user defined torque-deflection profiles [33], which we are working on integrating into our SEAs [34].

#### ACKNOWLEDGEMENTS

We thank D. Matten and K.A. Geberth for help with actuator design and EtherCAT communication, L. Hayhurst for help with robot fabrication, T. Xu for help with sensor design, and R. Desai for help with early experiments.

#### REFERENCES

[1] M.S. Redfern and T. Schumann. A model of foot placement during gait. *J Biomech*, 27(11):1339-1346, 1994.  
 [2] M. Raibert. Legged robots that balance. *MIT Press*, 1986.

[3] M.A. Townsend. Biped gait stabilization via foot placement. *J Biomech*, 18(1):21-38, 1985.  
 [4] S. Kajita and K. Tani. Study of dynamic biped locomotion on rugged terrain - derivation and application of the linear inverted pendulum mode. *Proc IEEE ICRA*, 1405-1411, 1991.  
 [5] R. Blickhan. The spring-mass model for running and hopping. *J Biomech* 22(11-12):1217-1227, 1989.  
 [6] A. Seyfarth *et al.* Swing-leg retraction: A simple model for stable running. *J Exp Bio*, 206(15):2547-2555, 2003.  
 [7] S. Kajita *et al.* Biped walking pattern generation by using preview control of zero-moment point. *Proc IEEE ICRA*, 1620-1626, 2003.  
 [8] J.E. Pratt *et al.* Capture point: A step toward humanoid push recovery. *Proc IEEE-RAS Humanoids*, 200-207, 2006.  
 [9] J.E. Pratt *et al.* Capturability-based analysis and control of legged locomotion, part 2: Application to M2V2, a lower-body humanoid. *IJRR*, 31(10):1117-1133, 2012.  
 [10] J. Urata *et al.* Online walking pattern generation for push recovery and minimum delay to commanded change of direction and speed. *Proc IEEE/RSJ IROS*, 3411-3416, 2012.  
 [11] J.A. Blaya and H. Herr. Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait. *IEEE Trans N Syst Rehab Eng*, 12(1):968-977, 2004.  
 [12] F. Sup *et al.* Upslope walking with a powered knee and ankle prosthesis: initial results with an amputee subject. *IEEE Trans N Syst Rehab Eng*, 19(1):71-78, 2011.  
 [13] J. Hitt *et al.* Dynamically controlled ankle-foot orthosis (dco) with regenerative kinematics: incrementally attaining user portability. *Proc IEEE ICRA*, 1541-1546, 2007.  
 [14] P.D. Neuhäus *et al.* Design and evaluation of MINA: A robotic orthosis for paraplegics. *Proc IEEE ICORR*, 1-8, 2011.  
 [15] S.K. Au *et al.* Powered ankle-foot prosthesis for the improvement of amputee ambulation. *Proc IEEE EMBS*, 3020-3026, 2006.  
 [16] B.E. Lawson *et al.* Stumble detection and classification for an intelligent transfemoral prosthesis. *Proc IEEE EMBS*, 511-514, 2010.  
 [17] B.E. Lawson. Control methodologies for powered prosthetic interventions in unilateral and bilateral transfemoral amputees. *PhD Thesis, VU*, 2014.  
 [18] M. Raibert. Hopping in legged systems - modeling and simulation for the two dimensional one-legged case. *IEEE Trans Syst, Man, Cyb, SMC-14(3)*:451-463, 1984.  
 [19] A.E. Dittrich *et al.* Obstacle avoidance in a simple hopping robot. *Proc CLAWAR*, 2006.  
 [20] M.F. Eilenberg *et al.* Control of a powered ankle-foot prosthesis based on a neuromuscular model. *IEEE Trans N Syst Rehab Eng*, 18(2):164-173, 2010.  
 [21] H. Geyer and H. Herr. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Trans N Syst Rehab Eng*, 18(3):263-273, 2010.  
 [22] R. Desai and H. Geyer. Robust swing leg placement under large disturbances. *Proc IEEE ROBOT*, 265-270, 2012.  
 [23] R. Desai and H. Geyer. Muscle-reflex control of robust swing leg placement. *Proc IEEE ICRA*, 2169-2174, 2013.  
 [24] S. Song *et al.* Integration of an adaptive swing control into a neuromuscular human walking model. *Proc IEEE EMBS*, 4915-4918, 2013.  
 [25] K. Hunt and F. Crossley. Coefficient of restitution interpreted as damping in vibroimpact. *J App Mech*, 440-445, 1975.  
 [26] A. Schepelmann *et al.* Development of a testbed for robotic neuromuscular controllers. *Proc RSS*, 385-392, 2012.  
 [27] G.A. Pratt and M.M. Williamson. Series elastic actuators. *Proc IEEE/RSJ IROS*, 399-406, 1995.  
 [28] S.A. Migliore *et al.* Novel nonlinear elastic actuators for passively controlling robotic joint compliance. *J Mech Des*, 129(4):406-412, 2007.  
 [29] M.D. Taylor. A compact series elastic actuator for bipedal robots with human-like dynamic performance. *MS Thesis, CMU*, 2011.  
 [30] B. Siciliano *et al.* *Robotics: Modelling, Planning and Control*. Springer-Verlag, 2010.  
 [31] N. Hansen. CMA-ES MATLAB implementation (Version 3.61), 2012.  
 [32] J.J. Eng *et al.* Strategies for recovery from a trip in early and late swing during human walking. *Exp Brain Res*, 102(2):339-349, 1994.  
 [33] A. Schepelmann *et al.* Compact nonlinear springs with user defined torque-deflection profiles for series elastic actuators. *Proc IEEE ICRA*, 3411-3416, 2014.  
 [34] J. Austin *et al.* Control and evaluation of series elastic actuators with nonlinear rubber springs. *Proc IEEE/RSJ IROS*, 2015.