Orienting in Mid-air through Configuration Changes to Achieve a Rolling Landing for Reducing Impact after a Fall

Jeffrey T. Bingham[1], Jeongseok Lee[1], Ravi N. Haksar[2], Jun Ueda[2] and C. Karen Liu[1]

***Abstract*— With no initial angular momentum an ordinary house cat is capable of flipping over onto its feet in mid-air and landing safely after a fall. As the field of robotics advances and robots become more dynamic, control algorithms for landing safely from a long, intended fall will become more necessary. Here we present an algorithm that leverages nonholonomic trajectory planning inspired by the falling cat to orient an articulated robot through configuration changes to achieve a pose that reduces the impact at landing. The calculated impact pose results in minimal loss of energy through rolling, while maximizing the rolling time. In addition to orienting and rolling, our controller guides the system to behave like a damped spring-mass system to reduce the magnitude of contact forces. Our framework is general and is applicable to systems that can be modeled as a connected tree of rigid bodies. We illustrate the feasibility of the algorithm through simulation and physical experiments with a planar three-link robot.**

I. INTRODUCTION

When falling, an ordinary house cat has the remarkable ability to twist its body in mid-air to land on its feet. This behavior is made more extraordinary by the fact that the cat is able to do this without any initial angular velocity and using no external forces to orient itself. Despite the considerable advances in robotics, this feat of agility is beyond the capability of current hardware platforms. However, as articulated robots advance and become more dynamic, it will be necessary to have at least a modest ability to properly survive the impact of a fall. Here we present an algorithm that leverages nonholonomic trajectory planning inspired by the falling cat to orient an articulated robot by configuration changes to achieve a final pose that rolls and reduces the impact during landing. We define a pose as the combination of configuration (joint state) and orientation (attitude of the root body with respect to the inertial frame).

Beyond orienting we also address the impact of landing, because, as the saying goes, "It's not the fall that kills you. It's the sudden stop at the end." The key word is *sudden* in this aphorism. The damage incurred by an impact with the ground is largely due to the magnitude of the impulse required to change the momentum of the falling body. Therefore, we suggest two strategies to reduce the detrimental effects of "the sudden stop at the end." First, perform a rolling action to reduce the change in momentum caused by the impact. Second, lengthen the duration of the impulse by acting like a damped spring-mass system, which decreases the magnitude of the impulse force at peak. To achieve these two goals on an articulated robot, we propose algorithms that compute the optimal impact pose and a set of joint stiffnesses to reduce the change in momentum and peak contact force during landing.

In this work we provide a theoretical framework for finding an impact pose and joint stiffness that reduce the impact impulse, as well as the sequence of configurations to orient an articulated robot to the impact pose during falling. The theoretical framework is general and should be applicable to any system that can be modeled as a connected tree of rigid bodies. Next, we present an application of this framework to a planar three-link robot with implementation details. Finally, we present simulation and experimental results to show the feasibility of our control algorithm.

II. RELATED WORK

Orienting through configuration change is greatly inspired by the motion of a falling cat. The physics of the falling cat has been studied in depth [1]. These same principles were used to explain humans flipping when diving [2], astronauts orienting in space [3] and more recently, lizards orienting while leaping [4]. The physical principles governing this phenomenon have also been generalized for any deformable body [5], [6], [7]. Optimal control has been proposed to solve for trajectories that minimize joint path length for a given orientation [8], [9], [7]. Also, recent work by the Full group has generated controllers for orienting when angular momentum is not initially zero [10], [11], [12]. Inspired by Hatton and Choset [13], [14], our work exploits the underlying geometry of the nonholonomic constraint of conserved angular momentum to solve for trajectories that allow orienting even when angular momentum is zero. We present our methods in terms of vector calculus, but other techniques, such as geometric phase and differential geometry, may offer more elegant solutions for the versed reader [13], [15].

Surviving impact after a fall is an important topic in robotics, but most research has focused on short, unintended falls. Previous work has used machine learning techniques to predict falling [16], manual hand tuning to design fall sequences [17], as well as an abstract model to control a safe fall [18], [19], [20]. When a fall is intended, such as landing from a jump, human landing behavior is observed to be similar to a damped spring-mass system [21]. This behavior has been described with good effect with the spring-loaded inverted pendulum (SLIP) model and the model has been applied to a multitude of robotic applications [22], [23]. Also, peak impact force has been shown to be reduced by

[1]School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA karenliu@cc.gatech.edu

[2]Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA

rolling after impact from experiential evidence in the sport of parkour (free-running) [24] and computer simulation [25], [26]. We extend previous work by exploring falls from height and incorporate both rolling and impact absorption with a "soft roll" on landing. Here our "soft roll" is a pose that maximizes rolling while also allowing the system to behave as a damped spring-mass system. Our work is most similar to Ha's physical simulation of falling and landing motion for animated characters [26]. In contrast to their work, we develop control algorithms that can be implemented in a physical system with a proof of concept on a robotic platform. Furthermore, we specifically address the case of orienting when initial angular momentum is zero and propose optimality criteria for landing in a soft, rolling pose.

III. CONTROL ALGORITHM

We propose a control algorithm for orienting an articulated robot during a fall using a series of configuration changes to reach a pose, that on impact, will result in the smallest change in momentum. Further, we propose control during the impact to transition into a rolling configuration and minimize peak contact forces by computing joint torques that drive the articulated body to behave as a damped spring-mass system.

To develop this control algorithm we make the following assumptions:

- Standard rigid body assumptions.
- Aerodynamics are negligible.
- Optimal rolling shape is a circle.
- Collision occurs at a single point.
- Collision is completely plastic.
- Angular momentum is conserved at impact.

Using these assumptions, we propose a control algorithm which consists of two iterative processes (Fig. 1). Based on the initial state of the robot, we first solve for the impact pose that minimizes the loss of kinetic energy due to impact and maximizes the rolling time. Next, we solve for a sequence of configuration changes that will drive the system to the desired impact pose. When the robot starts to fall, it is commanded to follow the planned configuration sequence. If the robot configuration deviates too much from the planned configuration, we re-solve for another sequence. This process repeats until the robot is in contact with the ground. During each time step of the landing phase, torques are solved to produce a damped-spring-like behavior at the center-of-mass and drive the system into a circular configuration. The entire process terminates when the robot makes the second contact with the ground.

```
Solve for impact pose
Solve orienting sequence
while falling do
    Track sequence
    if Tracking error beyond threshold then
        Re-solve orienting sequence
    end if
end while
while landing do
    Solve joint torques to match damped spring-mass and transform to rolling configuration
    Apply torques
end while
```

Fig. 1. Pseudo-code outline of how the control policy is achieved.

A. Optimizing impact pose

Given the initial kinematic state of the robot and a set of allowable contact points, $\mathcal{U}$, we optimize for an impact pose, $\mathbf{q}_a$, by minimizing the amount of energy lost during the impact and maximizing the rolling time to the next predicted contact. At the instant of impact, the system can be modeled as a single rigid body with locked joints impacting an impenetrable half-plane. Therefore, we define an inertial frame centered at the point of collision and a rigid body frame located at the center of mass of the rigid body by a vector, $\mathbf{c}$, and a rotation matrix, $\mathbf{R}$. The rate of change of rotation is defined with an angular velocity vector, $\boldsymbol{\omega}$, and the relation, $\dot{\mathbf{R}} = [\boldsymbol{\omega}]_\times \mathbf{R}$, where $[\cdot]_\times$ denotes the skew-symmetric matrix form of the cross product. Any local point on the rigid body, $\mathbf{u}$, is related to a global position, $\mathbf{x}$ by, $\mathbf{x} = \mathbf{R}\mathbf{u} + \mathbf{c}$. Finally, we can define the kinematics of a point on the rigid body, $\mathbf{x}_a$, where the rigid body contacts the ground as:

$$\mathbf{x}_a = \mathbf{R}\mathbf{u}_a + \mathbf{c} \tag{1}$$

$$\dot{\mathbf{x}}_a = \dot{\mathbf{R}}\mathbf{u}_a + \dot{\mathbf{c}} = [\boldsymbol{\omega}]_\times \mathbf{R}\mathbf{u}_a + \dot{\mathbf{c}} \tag{2}$$

Next, we define the linear, $\mathbf{L}$, and angular, $\mathbf{H}_a$, momentum about the contact point, where the rigid body has mass, m, and the moment of inertia about its center of mass of $\mathbf{I}_c$:

$$\mathbf{L} = m\dot{\mathbf{c}} \tag{3}$$

$$\mathbf{H}_a = \mathbf{R}\mathbf{I}_c\mathbf{R}^T\boldsymbol{\omega} + (\mathbf{c} \times \mathbf{L}) \tag{4}$$

Then, we recall the impulse and momentum equations with external forces, $\mathbf{F}$, and moments, $\mathbf{M}_a$, as:

$$\mathbf{L}_2 - \mathbf{L}_1 = \int_{t_1}^{t_2} \sum \mathbf{F} dt \tag{5}$$

$$\mathbf{H}_2 - \mathbf{H}_1 = \int_{t_1}^{t_2} \sum \mathbf{M}_a dt \tag{6}$$

We define the time period such that t_1 is the instant right before impact, such contact is made but no force has been transmitted to the rigid body; and the instant t_2 immediately after the impulse has been applied. If the time period $\epsilon = t_2 - t_1$ is very short and the impact is plastic, such that no rotation or translation occurs ($\mathbf{R}_1 \approx \mathbf{R}_2$ and $\mathbf{c}_1 \approx \mathbf{c}_2$), we find that angular momentum is conserved, even though linear momentum will be different before and after impact:

$$\begin{aligned} \mathbf{H}_2 - \mathbf{H}_1 &= \int_{t_1}^{t_2} (-\mathbf{R}\mathbf{u}_a \times m\mathbf{g})\, dt \\ &= (-\mathbf{R}\mathbf{u}_a \times m\mathbf{g})\,(t_2 - t_1) \approx \mathbf{0} \end{aligned} \tag{7}$$

$$\mathbf{H}_2 \approx \mathbf{H}_1 \tag{8}$$

The kinematics of the contact point after the impact are then defined to be:

$$\mathbf{c} = -\mathbf{R}\mathbf{u}_a \tag{9}$$

$$\dot{\mathbf{x}}_{a_2} = \mathbf{0} \tag{10}$$

Using (2) and (10) we arrive at a relation for the linear velocity of the center-of-mass after impact:

$$\dot{\mathbf{c}}_2 = -[\boldsymbol{\omega}_2]_\times \mathbf{R}\mathbf{u}_a \tag{11}$$

By substituting (11) and (9) into the momentum equation (4) we can solve for the angular velocity after impact:

$$\begin{aligned}
&\mathbf{R}\mathbf{I}_c\mathbf{R}^T\boldsymbol{\omega}_2 + (\mathbf{c} \times m\dot{\mathbf{c}}_2) = \mathbf{H}_1 \\
&\mathbf{R}\mathbf{I}_c\mathbf{R}^T\boldsymbol{\omega}_2 - (\mathbf{c} \times m[\boldsymbol{\omega}_2]_\times \mathbf{R}\mathbf{u}_a) = \mathbf{H}_1 \\
&\mathbf{R}\mathbf{I}_c\mathbf{R}^T\boldsymbol{\omega}_2 - (\mathbf{c} \times m\left(\boldsymbol{\omega}_2 \times \mathbf{R}\mathbf{u}_a\right)) = \mathbf{H}_1 \\
&\mathbf{R}\mathbf{I}_c\mathbf{R}^T\boldsymbol{\omega}_2 + (\mathbf{c} \times m\left(\mathbf{R}\mathbf{u}_a \times \boldsymbol{\omega}_2\right)) = \mathbf{H}_1 \\
&\mathbf{R}\mathbf{I}_c\mathbf{R}^T\boldsymbol{\omega}_2 + m\left[\mathbf{c}\right]_\times \left[\mathbf{R}\mathbf{u}_a\right]_\times \boldsymbol{\omega}_2 = \mathbf{H}_1 \\
&\mathbf{R}\mathbf{I}_c\mathbf{R}^T\boldsymbol{\omega}_2 - m\left[\mathbf{c}\right]_\times \left[\mathbf{c}\right]_\times \boldsymbol{\omega}_2 = \mathbf{H}_1
\end{aligned} \tag{12}$$

$$\boldsymbol{\omega}_2 = \left(\mathbf{R}\mathbf{I}_c\mathbf{R}^T - m\left[\mathbf{c}\right]_\times \left[\mathbf{c}\right]_\times\right)^{-1} \mathbf{H}_1 \tag{13}$$

Equation (13) shows that the angular velocity after impact can be expressed by the impact pose $\mathbf{q}_a$ and the angular momentum before the impact:

$$\mathbf{H}_1 = \mathbf{R}\mathbf{I}_c\mathbf{R}^T\boldsymbol{\omega}_1 + \mathbf{c}(\mathbf{q}_a) \times \mathbf{L}_1 \tag{14}$$

Because $\mathbf{R}\mathbf{I}_c\mathbf{R}^T\boldsymbol{\omega}_1$ and $\mathbf{L}_1$ are determined by the initial state, the only variable in $\mathbf{H}_1$ is $\mathbf{q}_a$. Therefore, we formulate the following optimization that solves for the impact pose $\mathbf{q}_a$ and the first contact point $\mathbf{u}_a \in \mathcal{U}$:

$$\min_{\mathbf{q_a},\, \mathbf{u}_a} \left(\eta\left(T_1 - T_2\right) - \left(1 - \eta\right) t_{roll}\right) \tag{15}$$

where the first term of the cost function minimizes the change in kinetic energy, T_i, and the second term maximizes the rolling time, t_{roll}. A trade-off is defined between these two objectives with a weighting parameter, η, arbitrarily set at 0.5.

Here T_1 is a constant determined by the initial conditions and T_2 is a function of the configuration of the robot and the allowable contact points:

$$T_i = \frac{1}{2}\left(m\dot{\mathbf{c}}_i^T\dot{\mathbf{c}}_i + \boldsymbol{\omega}_i^T\mathbf{R}\mathbf{I}_c\mathbf{R}^T\boldsymbol{\omega}_i\right) \tag{16}$$

The rolling time t_{roll} is defined as the elapsed time between the first contact $\mathbf{u}_a$ and the next one $\mathbf{u}_b$. To predict the next contact, we approximate the time that takes each allowable point to hit the ground by dividing its vertical distance to the ground by its vertical velocity after impact. The point with the shortest time is the predicted next contact $\mathbf{u}_b$.

$$t_{roll} = \min_{\mathbf{u}_b} \frac{-\left(\mathbf{R}\mathbf{u}_b - \mathbf{R}\mathbf{u}_a\right) \cdot \mathbf{n}}{\left([\boldsymbol{\omega}_2]_\times \mathbf{R}\mathbf{u}_b\right) \cdot \mathbf{n}} \tag{17}$$

In our implementation, we simplified the problem by assuming a fully extended joint configuration at impact and only solve for the orientation of the root. This simplification significantly improves the speed of the computation and is consistent with the intuition that making contact as early as possible provides longer "cushion time" to absorb impact.

B. Orienting through configuration

Given an initial pose $\mathbf{q}_o$ and the desired impact pose $\mathbf{q}_a$, we formulate an optimization to solve for a sequence of configurations. We begin with an articulated rigid body system described as a collection of links in a tree structure located with respect to a root body. Next we define the angular momentum with respect to the system's center-of-mass, $\mathbf{H}_c(\mathbf{q},\, \dot{\mathbf{q}})$, as a function of the generalized coordinates, q_k and speeds, $\dot{q}_k$. The angular momentum is linear with respect to the generalized speeds, so we can define a Jacobian, $\mathbf{J}_H(\mathbf{q})$, that is only a function of the generalized coordinates and relates the partial derivatives of the angular momentum with the generalized speeds:

$$\mathbf{H}_c = \sum_{k=1}^{p} \frac{\partial \mathbf{H}_c}{\partial \dot{q}_k}\dot{q}_k = \mathbf{J}_H\dot{\mathbf{q}} \tag{18}$$

This Jacobian may be partitioned into components pertaining to generalized speeds that orient the root body, $\dot{\mathbf{q}}^r$, and the rest of the degrees of freedom in the system, $\dot{\mathbf{q}}^s$, including those associated with configuration and root body translation:

$$\mathbf{H}_c = \mathbf{J}_H^r\dot{\mathbf{q}}^r + \mathbf{J}_H^s\dot{\mathbf{q}}^s \tag{19}$$

Using equation (19) we can express the generalized speeds of the root orientation in terms of the generalized speeds of the rest of the system:

$$\dot{\mathbf{q}}^r = -\left(\mathbf{J}_H^r\right)^{-1}\left(\mathbf{J}_H^s\dot{\mathbf{q}}^s - \mathbf{H}_c\right) \tag{20}$$

Furthermore, it is found that the resulting Jacobians of the generalized angular momentum are not dependent on the root body generalized coordinates. This is as expected for angular momentum about the center-of-mass, which should not be dependent on the location or orientation of the system of rigid bodies. Using this, we define a function associated with the change in angular momentum from a shape change:

$$\boldsymbol{\Gamma}(\mathbf{q}^s) = -\left(\mathbf{J}_H^r\right)^{-1}\mathbf{J}_H^s \tag{21}$$

Now if we multiply (20) through by dt we find the change in root body orientation:

$$d\mathbf{q}^r = \boldsymbol{\Gamma}(\mathbf{q}^s)d\mathbf{q}^s + \left(\mathbf{J}_H^r\right)^{-1}\mathbf{H}_c dt \tag{22}$$

If the initial angular momentum about the center-of-mass is zero, the second term of (22) vanishes. In the cases when the initial angular moment is nonzero, we define the quantity associated with the time-dependent component as the amount of "drift" caused by any initial angular momentum over an interval of time, $[t_o, t_a)$:

$$\mathbf{D}(\mathbf{q}^s) = \int_{t_o}^{t_a} \left(\mathbf{J}_H^r\right)^{-1}\mathbf{H}_c dt \tag{23}$$

An infinite set of configuration trajectories, $\mathbf{q}^s$, exist for a given change in orientation, $\mathbf{q}_a^r - \mathbf{q}_o^r$. To reduce the solution space, we formulate an optimization to solve for the minimal path length for the configuration variables that achieves the

desired root orientation with a constraint on the configuration velocities.

$$\begin{aligned} &\min_{\mathbf{q}^s} \int_{\mathbf{q}^s} d\mathbf{q}^s \\ &\text{subject to} \quad \mathbf{q}_a^r - \mathbf{q}_o^r = \int_{\mathbf{q}^s} \boldsymbol{\Gamma}(\mathbf{q}^s) d\mathbf{q}^s + \mathbf{D}(\mathbf{q}^s) \\ &\qquad\qquad \dot{\mathbf{q}}^s = v(t) \end{aligned} \tag{24}$$

The time interval associated with the drift term is an additional unknown that must be solved using the constraint on joint velocities. A local solution to the problem (24) can be found by first setting the drift term (23) to zero. The trajectory from the drift free solution and the constraint on velocity can be used to compute the time interval of drift and the amount of orientation change from this drift. Then, drift can be added as a constant term to the optimization and the optimization re-solved. Through iteration a final trajectory is solved.

C. Absorbing impact during collision

We now present a method to absorb contact force after the initial impact while changing into a configuration that is most likely to roll. We make the assumption that if our system changes into a circular configuration and behaves as a damped spring-mass on impact, the change in linear momentum can be absorbed with a smaller contact force compared to maintaining the landing configuration. For the damped spring-mass system we assume a Voigt element, a spring, k, in parallel with a damper, b, rigidly attached to an inertial reference at one end and a mass, m, in a gravitational field at the other end:

$$m\ddot{c}_z = -b\dot{c}_z - k(c_z - z_o) - mg \tag{25}$$

Given a step change in the velocity initial condition, an over damped solution will give the smallest peak force in the Voigt element. Assuming that contact velocity is known, the values for the spring and damper can be selected for an over-damped solution that nearly minimizes the peak contact force. To show this, we use arguments about the events that occur. When contact first occurs ($t = 0$) we have $c_z(t = 0) = z_o$ and $\dot{c}_z(t = 0) = v_o$, so the initial force is just the force in the damper, $F(t = 0) = -bv_o$. As time goes to infinity, the force in the damper will decay to zero and the only force will be what is left in the spring at maximum displacement, $F(t = \inf) = |kx_{max}| = |mg|$. Therefore, we can select the spring constant based on our desired maximum displacement:

$$k = \frac{mg}{x_{allowed}} \tag{26}$$

In the over-damped solution center-of-mass speed is monotonically decreasing in proportion to the force that is applied. Therefore, we want to exert the maximum amount of force at the very beginning, that is still less than the desired peak contact force. As such, we select the damping constant based on the final force magnitude:

$$b = \left| \frac{mg}{v_o} \right| \tag{27}$$

To implement this virtual behavior on the robot we solve for joint torques, $\boldsymbol{\tau}$, that attempt to match the center-of-mass kinematics of the damped spring-mass system while following a joint trajectory that drives the robot into a rolling configuration. The passive dynamics of the system cannot be altered, therefore to insure a controllable system we rewrite the generalized coordinates into the form, $\mathbf{q} = [\mathbf{q}^g \ \mathbf{q}^l]^T$; unactuated coordinates, $\mathbf{q}^g$; actuated coordinates, $\mathbf{q}^l$. This gives a form that uncovers the controllable states of the system. Then the equations are written with the terms: inertia matrix, $\mathbf{M}(\mathbf{q})$; coriolis and gravitational terms, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$; Jacobian for the contact point, $\mathbf{J}_a(\mathbf{q})$; and the contact force, $\mathbf{F}_a$. Note that here,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} + \mathbf{J}_a(\mathbf{q})^T \mathbf{F}_a \tag{28}$$

For the remainder of the relations the dependent variables are implied, e.g. $\mathbf{J}_a(\mathbf{q}) \equiv \mathbf{J}_a$. We wish to satisfy (25) and (28), while following a specified joint trajectory, $\bar{\mathbf{q}}^l$. Here we assume that the joint trajectory is given and results in a circular configuration designed for rolling. Then, we split the torque applied into a component that tracks the joint trajectory for rolling, $\boldsymbol{\tau}_r$ and another that maintains the center-of-mass trajectory, $\boldsymbol{\tau}_c$:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_r + \boldsymbol{\tau}_c \tag{29}$$

We define the joint trajectory following torque, $\boldsymbol{\tau}_r$, with proportional-derivative tracking:

$$\boldsymbol{\tau}_r = -\mathbf{K}_p\left(\mathbf{q}^l - \bar{\mathbf{q}}^l\right) - \mathbf{K}_d\dot{\mathbf{q}}^l \tag{30}$$

Next, we will formulate kinematic relations necessary to solve for the remaining torque that will most closely drive the system to match the damped spring-mass kinematics. We define a relations for the center-of-mass kinematics in terms of the generalized coordinates:

$$\dot{\mathbf{c}} = \mathbf{J}_c\dot{\mathbf{q}} \tag{31}$$

$$\ddot{\mathbf{c}} = \dot{\mathbf{J}}_c\dot{\mathbf{q}} + \mathbf{J}_c\ddot{\mathbf{q}} = \dot{\mathbf{J}}_c\dot{\mathbf{q}} + \mathbf{J}_c\mathbf{M}^{-1}(-\mathbf{C} + \boldsymbol{\tau} + \mathbf{J}_a^T\mathbf{F}_a) \tag{32}$$

We now formulate an optimization to solve for the remaining joint torque $\boldsymbol{\tau}_c$ which minimizes the difference between the center-of-mass acceleration and the desired acceleration following the damped spring-mass equation: $\ddot{c}_z = -\frac{b}{m}\dot{c}_z - \frac{k}{m}c_z - (g - \frac{k}{m}z_o)$.

$$\min_{\boldsymbol{\tau}_c} \frac{1}{2}\left([\ddot{\mathbf{c}}]_z - \ddot{c}_z\right)^T\left([\ddot{\mathbf{c}}]_z - \ddot{c}_z\right) \tag{33}$$

where $[\mathbf{v}]_z$ denotes the projection of a vector $\mathbf{v}$ onto the z-axis. To reduce the complexity of solving the contact force, $\mathbf{F}_a$, and center-of-mass acceleration, $\ddot{\mathbf{c}}$, simultaneously, we use the value of contact force from the previous measurement. This quadratic program, (33), is then solved at each measurement interval to generate the necessary torque to guide the system into a rolling configuration while behaving as a damped spring-mass system (25).

IV. METHODS

As a first step towards testing the applicability of our control algorithm we chose to explore a planar three-link robot as one of the simplest systems capable of orienting through configuration changes. Here we describe both the physical and virtual models of the robot. Then, we outline the physical and virtual experiments used to test the feasibility of the control algorithm for this specific robot.

A. Robot hardware

Our hardware platform consisted of a symmetric three-link robot and an air-table. The three-link robot allowed simple configuration changes and the air-table was used to restrict motion to a plane, control fall time, and minimize external forces on the robot. A full list of parts, CAD models and microprocessor code are available by contacting the authors.

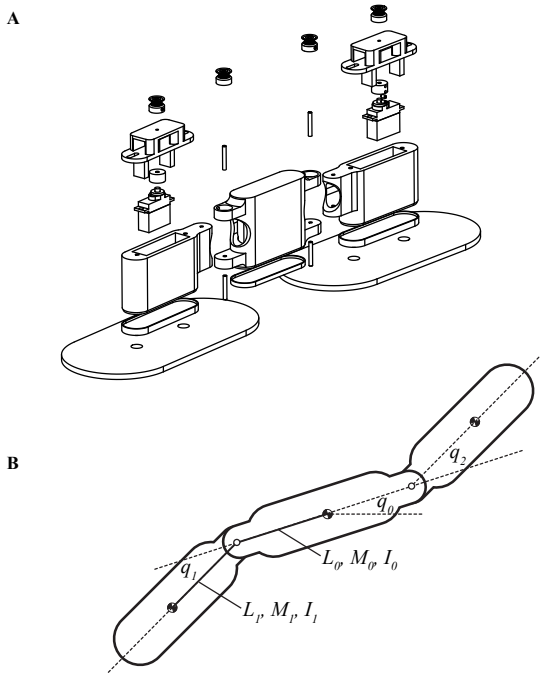

Fig. 2. A) Exploded view of three-link robot. Driving belt, battery pack and micro-circuitry are not shown. B) Definition of joint angles and parameters for simulation.

TABLE I
DIMENSIONS AND PARAMETERS OF THREE-LINK ROBOT

Parameter	Value	Units
L_0	5.60	cm
L_1	5.44	cm
M_0	210.2	g
M_1	120.2	g
I_0	1499.7	g-cm^2
I_1	895.1	g-cm^2

The three-link robot was designed in solid modeling software (Solidworks 2013, Dassault Systemes, Waltham, MA). The robot consisted of a central body with two symmetric legs connected by pin joints creating three planar links (Fig. 2). The structural components were printed in ABS plastic using a 3D printer (Dimension SST 768, Stratasys, Eden Prairie, MN). The bottom surface area of the robot was increased to 384 cm^2 with two "paddles" laser cut out of 0.5 cm thick acrylic sheet and attached to each leg. Actuation was achieved for each joint with hobby-grade mini-servos (HD-1705MG, Pololu, Las Vegas, NV) having stall torque of 2.0 N-cm and angle range of 100°. Power was supplied at 5 V with four AA NiMH rechargeable batteries. Remote control was implemented using an Arduino Pro Mini 328 and Bluetooth Mate Silver (Sparkfun, Boulder, CO).

An air-table was constructed to provide a near frictionless surface for the three-link robot to rotate and slide on. The air-table was built similar to conventional air-hockey tables, with a smooth surface and sealed air-box using melamine covered particle board. The testing surface was a rectangle 122 $\times$ 244 cm. The surface was treated with spray-able teflon coating and perforated with 0.16 cm diameter holes spaced 2.54 cm apart. The sealed air chamber was pressurized with an electric leaf blower (Toro Model 51609 Leaf Blower, Home Depot, Atlanta, GA) capable of 1.84×10^5 cm^3/s and exit velocity of 1.05×10^4 cm/s.

B. Simulated model

The simulated version of the robot was modeled to be as close to the physical model as possible with the differences of no air table and the collision geometry did not include the shape of the paddles. Parameters for the simulated model, including dimensions and mass properties, were taken from the solid model (Table I). Joint limits were set to ($q_1 = [0°,\ 90°]$ and $q_2 = [-90°,\ 0°]$). A description of the robot was written into a URDF (Unified Robot Description Format) file. Simulation was achieved using an open-source multi-body dynamics engine (DART 3.1, https://github.com/dartsim/dart) with control code written in C++ and optimization performed with NLopt [27]. The root body was given all three degrees-of-freedom for planar motion and joint torques were applied using proportional-derivative tracking control. Where appropriate, the robot simulation was operated with feedback of the full kinematic state and contact forces. On-line trajectory planning was computed by interpolating the optimal configuration path using a cubic interpolation scheme, since the optimal configuration path does not provide velocity information. Feedback was used to re-solve the orientation plan when measured root body rotation deviated from the plan by more than 5°. The model file and the source code for the controller are available upon request.

C. Design of experiments

Three types of experiments were performed both virtually in simulation and physically on the robot. All experiments were done with zero initial linear and angular velocity. The first experiment was a stationary orientation behavior, where the robot oriented to a specified pose without the effect of gravity. This was used to test the orientation planner portion

of the control algorithm. The second experiment was a dead drop of the robot with no control in a fixed configuration. This experiment was used as a baseline to compare against the algorithm. The final experiment was a drop of the robot while it oriented to the optimized impact pose, followed by a rolling motion. This was used as a test of the full algorithm.

1) Simulation: Simulation was run on a desktop computer (3.70 GHz, 16 GB RAM, Ubuntu 13.10) with an integration step of 0.1 ms. The stationary orienting experiment solved for joint trajectories to orient the root body by 20°while starting and ending in a straightened configuration. For the dead drop and falling experiments, conditions were set such that the drop height was 220 cm, acceleration was 0.157 m/s^2 (to match physical experiments), and configuration of $q_1 = q_2 = 0°$. The dead drop was performed with orientations of $q_0 = [10,\ 20,\ 30,\ 40,\ 50,\ 60,\ 70,\ 80]°$ and two cases were tested: one with the absorbing impact control on and another with the joints locked. The falling experiment used an initial orientation of $q_0 = 45°$ and the impact orientation (i.e. q_0 at impact) was found by solving the minimization (15) with the configuration at impact set to $q_1 = q_2 = 0°$. The rolling trajectory was set to transform to a U-shaped configuration where $q_1 = -90°$ and $q_2 = 90°$.

2) Physical: Robot movement was captured with a motion capture system (Vicon, Oxford, UK) utilizing 12 MX40+ cameras and Blade 1.5 software. A custom marker set of six 0.9 cm diameter reflective balls was used and motion data was sampled at 120 Hz. In addition, live video was recorded at 30 Hz using a point-and-shoot camera. Post-processing of the motion capture data was done in Matlab (Mathworks, Natick, MA) by applying a third-order, twenty-one sample wide savitsky-golay filter to the marker traces. Body translation, rotation and joint angles were then computed from body segments defined by the marker placements.

The robot was operated entirely open-loop. A fixed joint trajectory was programmed into the robot and triggered remotely. The joint trajectory was programmed as desired joint angles and delays between moves. The impact orientation and the open loop configuration sequence were solved using the simulation framework. The optimal configuration sequence generated the maximum rotation of the robot for a closed cycle where the joint angles started and ended at 0°. Each experiment was repeated five times. The stationary orienting experiments were run with the robot in the middle of the air-table surface and the closed cycle was repeated four times to generate rotation of the root body. For the falling experiments the air-table was slightly tilted to produce a gravitational acceleration, the robot was released in a straightened configuration with the root body at a 45°angle with the collision surface, and allowed to fall approximately 220 cm. Acceleration was estimated from the dead drop experiments using the length of travel and time of the fall. In the falling orientation experiment the robot cycled 4 times and then moved to a U shaped configuration where $q_1 = -90°$ and $q_2 = 90°$. Angular momentum was computed for the experimental trials using the recorded motion capture data. Components of the root body rotation induced by configuration change and angular momentum change were then calculated from the computed angular momentum.

V. RESULTS

A. Simulation

The maximum rotation within the joint limits was found to be a cycle of configurations that moved the robot in an I-C-L sequence, where the letters of the alphabet are used as cartoons for the robot configuration. The rotation of the root body for a single maximum orientation cycle was 4.60°when the robot closely tracked the trajectory (Figure 3A). For the static orientation the algorithm required approximately 4.35 cycles to rotate the root body 20°.

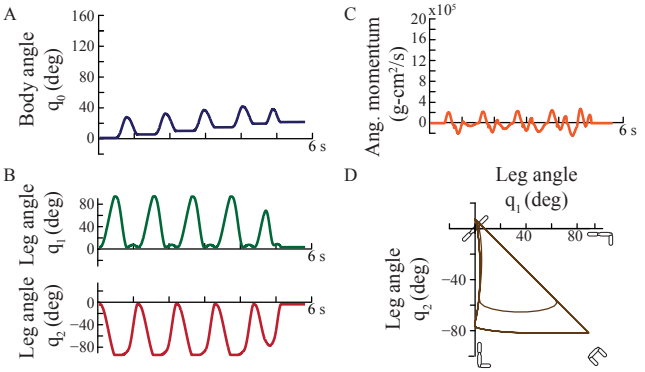

Fig. 3. Simulation results for stationary orienting with A) total orientation angle B) joint angles C) angular momentum about the center-of-mass D) configuration trajectory of joint angles.

In the dead drop experiments, the initial impact resulted in peak contact forces that were higher when the angle between the robot and the ground increased (Figure 4). Conversely, the peak contact force decreased for the second impact as initial impact angle increased. For the initial impact, peak ground reaction force was found to be the same between the cases of locked joints and active impact absorbing control. However, active control resulted in lower peak contact forces for the second impact.

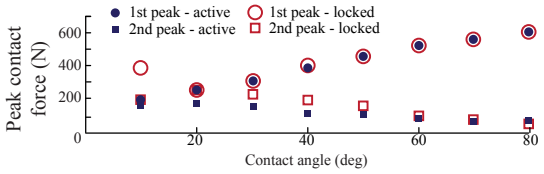

Fig. 4. Peak contact force decreases as angle of impact decreases. Trials with locked joints (open data points) exhibited slightly greater contact forces over those with active control (solid data points) for the second impact.

For the falling experiment (Figures 5 and 6) the impact angle that minimized the cost function (15) was found to be 75°. The planning optimization took 2.86 s and no re-planning was required during the falling phase, as maximum deviation from the plan was only 2.06°. During the landing phase, each iteration of the torque optimization required approximately 0.23 ms.

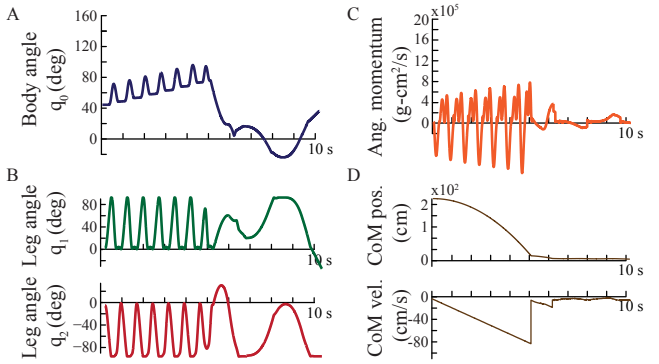

Fig. 5. Simulation results for orienting while falling with A) total orientation angle B) joint angles C) angular momentum about the center-of-mass D) center-of-mass vertical position and velocity. Impact occurs at approximately 5 s.

Fig. 6. Screen capture of orienting while falling sequence from simulation. Breaks in base mark different frames. Sequence starts at last configuration change in falling phase and goes through landing phase.

B. Experiments

The average rotation of the root body for a single cycle during the stationary orienting experiments was $6.2 \pm 1.3°$. Due to external perturbations from the air-table only $2.3 \pm 0.2°$ of rotation was attributed to configuration change (dashed line of Figure 7). This was calculated by solving for the expected theoretical orientation (22) based on the measured joint angles. Across the stationary orienting trials the total rotation over four cycles was $25.5 \pm 3.0°$ with $8.1 \pm 1.1°$ attributed to the change in configuration (Fig. 7). From the dead-drop trials the average drop acceleration was found to be 15.8 ± 0.4 cm/s^2. Falling orientation was found to have similar cycle performance to the stationary trials. Also, for the exemplary orienting while falling trial, shown in Fig. 8, the maximum change in angular momentum from external perturbations was 2×10^6 g-cm^2/s.

VI. CONCLUSIONS AND FUTURE WORK

We presented theory and proof of concept for a control algorithm capable of orienting a falling articulated robot into a landing pose that reduces the effects of the impact. The current algorithm has three major limitations. The optimal configuration path solved by (24) did not take into account the dynamics of the robot. Additional treatment is needed to ensure that the desired joint trajectories are feasible under the dynamic constraints of the robot. Second, our implementation of the spring-mass behavior on landing did not achieve the desired result of lengthening time of the impulse. This was in part due to the inability to simulate the collision event at a high-enough time resolution, and points out flaws in the ability to implement this particular portion of the algorithm in hardware. Finally, the rolling pose was arbitrarily designed based on the assumption that the optimal rolling shape is a circle. There is no theoretical support that the U-shaped pose is the optimal choice.

For future directions, we are interested in applying this control algorithm to more complex articulated rigid body systems with 3D orientation at the root. We expect that a new optimization formulation or some pre-computation is required to achieve real-time computation. Investigating different rolling poses and their impacts on contact forces can also be a fruitful future direction.

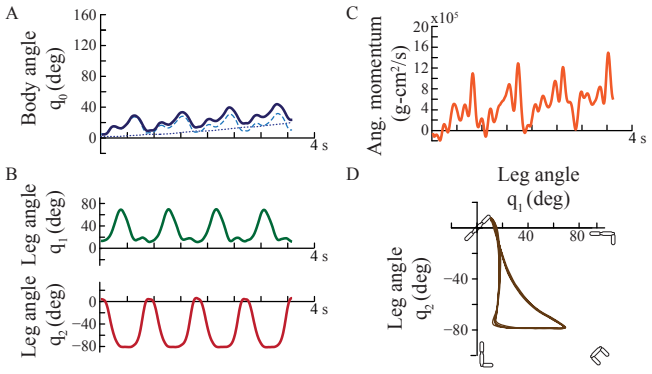

Fig. 7. An exemplary stationary orienting trial with A) total orientation angle (solid), angle due to drift (dotted), angle due to configuration change (dashed) B) joint angles C) angular momentum about the center-of-mass D) joint angle cycle.

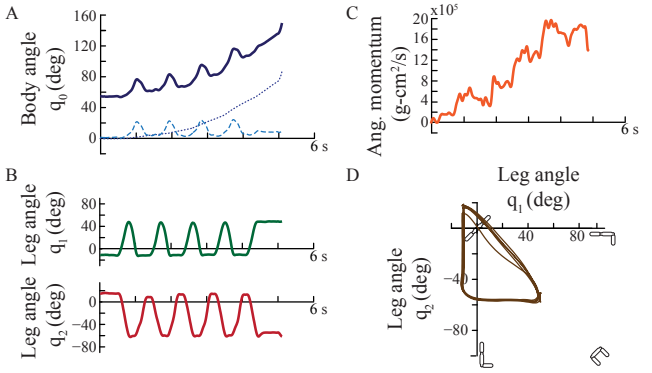

Fig. 8. An exemplary orienting while falling trial with A) total orientation angle (solid), angle due to drift (dotted), angle due to configuration change (dashed) B) joint angles C) angular momentum about the center-of-mass D) joint angle cycle.

ACKNOWLEDGMENT

The authors would like to acknowledge the thoughtful discussion and insight from Nathan Bunderson, Jie Tan, Sehoon Ha and Yunfei Bai. This work was generously funded in part by NSF EFRI-1137229 and DARPA OSRF-12-007.

REFERENCES

[1] T. Kane and M. Scher, "A dynamical explanation of the falling cat phenomenon," *International Journal of Solids and Structures*, vol. 5, no. 7, pp. 663–IN2, Jul. 1969.

[2] C. Frohlich, "Do springboard divers violate angular momentum conservation?" *American Journal of Physics*, vol. 47, no. 7, p. 583, 1979.

[3] T. Kane and M. Scher, "Human self-rotation by means of limb movements," *Journal of Biomechanics*, vol. 3, no. 1, pp. 39–49, Jan. 1970.

[4] A. Jusufi, D. T. Kawano, T. Libby, and R. J. Full, "Righting and turning in mid-air using appendage inertia: reptile tails, analytical models and bio-inspired robots." *Bioinspiration & biomimetics*, vol. 5, no. 4, p. 045001, Dec. 2010.

[5] R. Montgomery, "Gauge theory of the falling cat," *Fields Inst. Communications*, vol. 1, pp. 193–218, 1993.

[6] A. Shapere and F. Wilczek, "Gauge Kinematics of Deformable Bodies," Tech. Rep., 1988.

[7] R. W. Batterman, "Falling cats, parallel parking, and polarized light," *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, vol. 34, no. 4, pp. 527–557, Dec. 2003.

[8] R. Montgomery, "Nonholonomic control and gauge theory," in *Nonholonomic Motion Planning*. Kluwer Academic Publishers, 1993, ch. Nonholonom, pp. 343–377.

[9] ——, "Optimal Control of Deformable Bodies and Its Relation to Gauge Theory," in *The Geometry of Hamiltonian Systems*, 1991, pp. 403–438.

[10] E. Chang-Siu, T. Libby, M. Brown, R. J. Full, and M. Tomizuka, "A nonlinear feedback controller for aerial self-righting by a tailed robot," in *2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany: IEEE, May 2013, pp. 32–39.

[11] E. Chang-Siu, T. Libby, M. Tomizuka, and R. J. Full, "A lizard-inspired active tail enables rapid maneuvers and dynamic stabilization in a terrestrial robot," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Sep. 2011, pp. 1887–1894.

[12] T. Libby, T. Y. Moore, E. Chang-Siu, D. Li, D. J. Cohen, A. Jusufi, and R. J. Full, "Tail-assisted pitch control in lizards, robots and dinosaurs." *Nature*, vol. 481, no. 7380, pp. 181–4, Jan. 2012.

[13] R. L. Hatton and H. Choset, "Geometric motion planning: The local connection, Stokes' theorem, and the importance of coordinate choice," *The International Journal of Robotics Research*, vol. 30, no. 8, pp. 988–1014, Jun. 2011.

[14] ——, "Connection vector fields and optimized coordinates for swimming systems at low and high reynolds numbers," in *ASME 2010 Dynamic Systems and Control Conference, Volume 1*. ASME, 2010, pp. 817–824.

[15] K. Crane, F. de Goes, M. Desbrun, and P. Schroder, "Digital geometry processing with discrete exterior calculus," in *ACM SIGGRAPH 2013 courses*, ser. SIGGRAPH '13. New York, NY, USA: ACM, 2013.

[16] S. Kalyanakrishnan and A. Goswami, "Learning to Predict Humanoid Fall," *International Journal of Humanoid Robotics*, vol. 08, no. 02, pp. 245–273, Jun. 2011.

[17] J. Ruiz-del Solar, R. Palma-Amestoy, R. Marchant, I. Parra-Tsunekawa, and P. Zegers, "Learning to fall: Designing low damage fall sequences for humanoid soccer robots," pp. 796–807, 2009.

[18] "Safe fall: Humanoid robot fall direction change through intelligent stepping and inertia shaping," *2009 IEEE International Conference on Robotics and Automation*, 2009.

[19] K. Fujiwara, S. Kajita, K. Harada, K. Kaneko, M. Morisawa, F. Kanehiro, S. Nakaoka, and H. Hirukawa, "An optimal planning of falling motions of a humanoid robot," *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[20] F. Kanehiro, S. Kajita, K. Yokoi, H. Hirukawa, and K. Kaneko, "UKEMI: Falling Motion Control to Minimize Damage to Biped," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, no. October, 2002, pp. 2521–2526.

[21] M. Santello, "Review of motor control mechanisms underlying impact absorption from falls." *Gait & posture*, vol. 21, no. 1, pp. 85–94, Jan. 2005.

[22] P. Holmes, R. Full, D. E. Koditschek, and J. Guckenheimer, "The dynamics of legged locomotion: models, analyses, and challenges," *SIAM Review*, vol. 48, no. 2, pp. 207–304, 2006.

[23] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behaviour explains basic dynamics of walking and running," *Proc Biol Sci*, vol. 273, no. 1603, pp. 2861–2867, 2006.

[24] D. Edwardes, *The Parkour and Freerunning Handbook*. It Books, 2009.

[25] D. F. Brown, A. Macchietto, K. Yin, and V. Zordan, "Control of rotational dynamics for ground behaviors," in *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '13*. New York, New York, USA: ACM Press, 2013, p. 55.

[26] S. Ha, Y. Ye, and C. K. Liu, "Falling and landing motion control for character animation," *ACM Transactions on Graphics*, vol. 31, no. 6, p. 1, Nov. 2012.

[27] S. G. Johnson, "The NLopt nonlinear-optimization package." [Online]. Available: http://ab-initio.mit.edu/nlopt