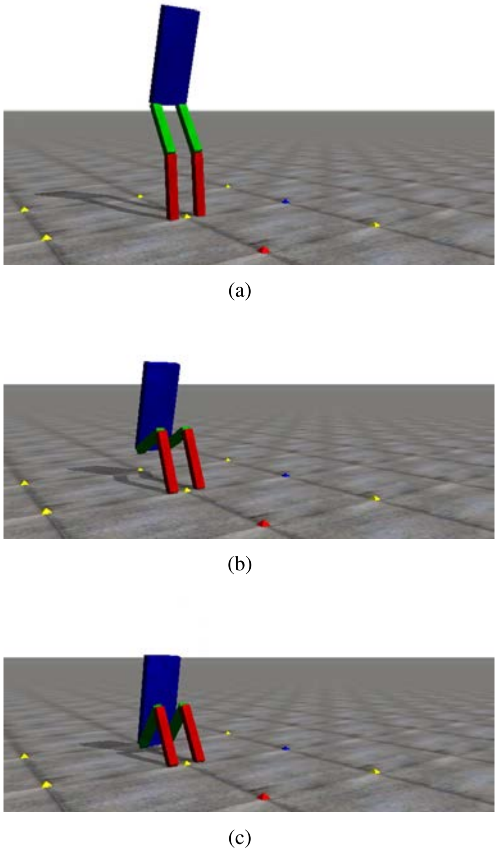
In Proc. American Control Conference (ACC'12), June, 2012.

Whole-Body Trajectory Optimization for Humanoid Falling

Jiuguang Wang, Eric C. Whitman and Mike Stilman

***Abstract*— We present an optimization-based control strategy for generating whole-body trajectories for humanoid robots in order to minimize damage due to falling. In this work, the falling problem is formulated using optimal control where we seek to minimize the impulse on impact with the ground, subject to the full-body dynamics and constraints of the robot in joint space. We extend previous work in this domain by numerically approximating the resulting optimal control, generating open-loop trajectories by solving an equivalent nonlinear programming problem. Compared to previous results in falling optimization, the proposed framework is extendable to more complex dynamic models and generate trajectories that are guaranteed to be physically feasible. These results are implemented in simulation using models of dynamically balancing humanoid robots in several experimental scenarios.**

***Index Terms*— Humanoid robot, optimal falling strategy, pseudospectral optimal control**



(a)

(b)

(c)

Fig. 1. Controlled fall: a three-link humanoid model falls backwards and lands at the hip joint. While falling, the torso bends forward until reaching the joint limit with respect to the thigh joint. At the final configuration, the torso is almost upright.

I. Introduction

Dynamically balancing humanoid robots are inherently unstable structures and any humanoid with sufficient physical capabilities poses a significant safety risk. On the other hand, humanoid robots are designed to imitate the manipulative, locomotive, perceptive, communicative, and cognitive abilities of humans [1] and are compatible with made-for-human tools and environments. Before we can expect to move these robots beyond the structured settings of research and into human populated environments, we must address the issue of robustness, particularly on how systems can be designed to gracefully cope with failures.

The robustness of autonomous robots in cluttered and uncontrolled human environments is a key area in the study of *physical human-robot interaction*. The ability to produce safe and controlled responses should a fault occur is one of the critical components that enable robots to effectively assist in physical tasks. In this work, we develop controllers for bipedal humanoid robots in the event of a fall as these systems are prone to falling when performing demanding locomotive movements. When left uncontrolled, falls can lead to catastrophic physical damage to the robot and its surroundings.

There is a vast collection of previous work in dynamic balancing strategies under disturbance for fall *avoidance*; for example, planning when and where to take a step [2] and feed-forward force control that achieves compliance [3]. In this work, we focus on the scenario where the balancing control has failed and a fall is inevitable. There are three main issues that naturally arise in this domain: *detecting* the fall (anomaly detection), choosing the optimal *direction* to fall (in order to ensure human safety), and choosing *how* to fall (in order to minimize damage to the robot). This work lies within the last category of damage minimization. Section II gives a summary of related work in each category and summarizes how they relate to the proposed approach.

Our design philosophy for falling control relies heavily on the ability of optimization to produce *emergent behaviors*. For a humanoid robot with many degrees-of-freedom, designing behaviors by hand is a difficult, if not intractable, task. While there have been attempts to use motion capture [4] to extract or transfer human motion to robot control in locomotive scenarios such as tripping recovery [5], the significant differences in the kinematic structures of humans

Jiuguang Wang and Eric Whitman are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania, 15213, USA. Emails: robot@cmu.edu, ewhitman@cmu.edu

Mike Stilman is with the Center for Robotics and Intelligent Machines at the Georgia Institute of Technology, Atlanta, Georgia, 30332, USA. Email: mstilman@cc.gatech.edu

and humanoid robots make this approach difficult to apply in practice. Instead, optimization is an effective tool by which we can generate behaviors from a system model using properly defined objective functions and constraints.

The primary contribution of this work is an optimization-based falling controller where the whole-body motion of the robot in joint space is automatically generated by defining appropriate optimization objectives. We aim to solve the optimal control problem numerically [6] with direct collocation using pseudospectral methods [7]. Specifically, a complex nonlinear optimal control problem is transcribed into an equivalent nonlinear programming problem (NLP) by parameterizing the state and control spaces using global polynomials and collocating the differential-algebraic equations using nodes obtained from a Gaussian quadrature. The resulting NLP can be solved by a numerical optimization solver to produce the locally optimal solution to the optimal control problem. We show that this approach is a natural choice for a nonlinear dynamic model with state and input constraints typical for a humanoid robot.

The remainder of this paper is organized as follows: Section II gives a summary of related work and reviews the state-of-the-art approaches to optimal falling control. Section III describes the dynamic model used in the optimization as well as the formulation of the optimization objective using an impact model. Section IV presents the pseudospectral method that transcribes the optimal control problem into an NLP, as well as the formation of the falling problem using an impulse-based optimization objective with corresponding dynamic constraints. Section V details the performance of the approach on simulated models. Finally, Section VI concludes the paper and gives future research directions.

II. Related Work

Fall detection for humanoid robot has been addressed in a number of previous studies, most notably in Renner [8], Karssen [9], and Kalyanakrishnan [10], where strategies for instability and fall detection were proposed using probabilistic and machine learning-based methods. In this work, we assume that the detection problem has been solved and the switch to falling control is done automatically.

For fall direction selection, Nagarajan [11] and Yun [12] have developed a class of approaches that utilize stepping strategies to change the bearing of the robot and falling to avoid multiple static obstacles in the environment. This line of research is primarily motivated by human safety and does not take into account damage minimization for the robot.

A number of strategies have proposed to minimize damage to the robot during falling, ranging from hand-designed behaviors to optimization generated controllers. The application of falling strategies is popular in the robot soccer domain, where falling occurs frequently due to the interactions of multiple robots in a small environment. Ruiz del Solar [13] utilized realistic simulation tools to hand-design fall sequences for soccer robots. A number of control designs based on human heuristic strategies have been proposed, for example, using techniques from martial arts [14] or defining desirable regions for impact such as the knee [15] or the backpack [16]. Optimization-based approaches were presented in Ogata [17] where a limited model based on the 3D inverted pendulum was used. More complex models can be found in Fujiwara [18] where variational techniques were used for optimizing joint trajectories.

We argue that an optimization-based approach encompasses heuristics-based techniques since appropriate heuristics can be embedded in the objective function to generate desired behaviors. The proposed approach is superior to previous optimization frameworks in several key aspects. First, we utilize a numerical approach for the optimal control problem which extends to more complex dynamic models that cannot be addressed by analytic methods. Second, we incorporate physical constraints on the robot (such as joint limits and actuator limits) into the optimization process, eliminating the need to use a separate viability check to discard trajectories that are physically infeasible. Finally, we introduce an alternative impact model based on the effective mass of the robot at impact, which is easier to model for more complex, multi-limbed humanoids.

III. Model

A. Overview

We model a standing humanoid robot in the sagittal plane as a three link rigid-body system. The planar diagram for the system is shown in Fig. 2, with the physical parameters listed in Table I. In this form, the three links represent the shank, thigh, and torso, respectively. We refer to the three actuated joints as the foot, knee, and hip joints, respectively. In this section, we derive the rigid-body dynamics for this three link system and the corresponding impact model to determine the impulse upon impact with the ground.

B. Rigid-body dynamics

To derive the rigid-body dynamics for the robot, we define the generalized coordinates of the system $\mathbf{q}(t) = (q_1(t), q_2(t), q_3(t))^T$ according to Fig. 2. We treat the three-link systems as a single kinematic chain and follow a standard derivation for manipulators, simplifying to a form that is linear in acceleration and torque

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \tau \tag{1}$$

where $\tau \in \mathbb{R}^m$ represents torques while $\mathbf{M} \in \mathbb{R}^{n \times n}$ and $\mathbf{N} \in \mathbb{R}^n$ are state-dependent matrices. $\mathbf{M}$ consists of coefficients on acceleration and is square, symmetric, and positive definite. $\mathbf{N}$ consists of all other terms including centripetal, Coriolis and gravitational forces. We can rewrite (1) as a series of first-order nonlinear equations in the control-affine form

$$\dot{\mathbf{x}}(t) = \mathbf{g}(\mathbf{x}) + \mathbf{w}(\mathbf{x})\mathbf{u}(t), \tag{2}$$

that is nonlinear in the states and linear in the control inputs, where $\mathbf{g}(\mathbf{x})$ and $\mathbf{w}(\mathbf{x})$ can be written as

$$\mathbf{g}(\mathbf{x}) = -\mathbf{M}^{-1}(\mathbf{q})\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) \tag{3}$$

$$\mathbf{w}(\mathbf{x}) = \mathbf{M}^{-1}(\mathbf{q}), \tag{4}$$

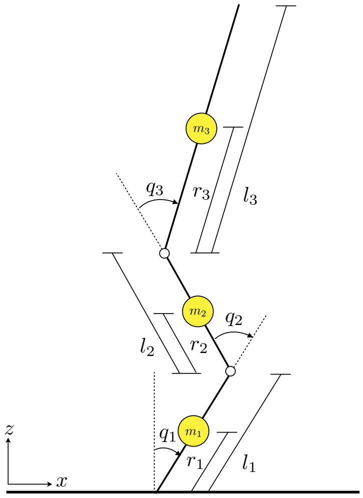


Fig. 2. Planar model of a legged humanoid robot with three links: shank, thigh, and torso. The three actuated joints q_i are the foot, knee, and hip defined in generalized coordinates. The parameters for the model is given in Table I.

TABLE I
PHYSICAL PARAMETERS

	m (kg)	r (m)	I_y $(kg \cdot m^2)$	l (m)
Shank (l1)	7.0	0.3	0.10	0.5
Thigh (l2)	17.0	0.3	0.30	0.5
Torso (l3)	53.5	0.35	3.0	0.8

We assume, without loss of generality, that the origin is an equilibrium of the system.

C. Impact dynamics

In controlled falling, the optimization objective is defined to provide some measure of "damage" upon impact with the ground. In this work, the optimization objective is quantified as the total impulse at impact, i.e., the change in momentum on the body of the robot as it hits the ground. Based on the geometry and the initial condition, either the head or the knee joint will hit the ground first in the forward falling case. For backwards falls, the hip will hit the ground first. In either case, we model the collision as inelastic and frictionless (consisting of a purely vertical force) then attempt to minimize the impulse imparted during the collision. Following an inelastic collision with the ground, the collision point will have zero vertical velocity. This means that we know the vertical change in velocity of the impact point during the collision, Δv_z, and we need only compute the effective mass in this configuration, $m(q)_{impact}$, to determine the total impulse.

We use the fact that the dynamics are linear in force to determine $m(q)_{impact}$. First we compute the vertical acceleration at impact resulting from only gravity, centripetal, and Coriolis forces, $a_{z,g}$. We then compute the same quantity but with a unit (1 N) vertical force at the impact point, $a_{z,f}$. The effective mass is then

$$m(q)_{impact} = \frac{1}{a_{z,f} - a_{z,g}}. \tag{5}$$

We can now calculate the impact impulse as

$$I = m(q)_{impact} \Delta v_z. \tag{6}$$

IV. Optimization

A. Overview

We formulate the falling control as an optimal control problem where the objective function is defined based on the impact dynamics of the fall. In this approach, we decompose the optimal control problem into an NLP by the pseudospectral Legendre method [19], a well-established approximation in trajectory optimization. Once the NLP description is obtained, we further transcribe it into a sequential quadratic programming (SQP) problem and solve it using standard numerical optimization solvers.

B. Optimal control formulation

Consider the finite-horizon optimal control problem where the objective function,

$$\begin{aligned} \min \quad \mathbf{J}(\mathbf{x}(\cdot), \mathbf{u}(\cdot), \tau_0, \tau_f) &= \mathbf{\Phi}(\mathbf{x}(\tau_0), \mathbf{x}(\tau_f), \tau_0, \tau_f) \\ &+ \int_{\tau_0}^{\tau_f} \mathbf{C}(\mathbf{x}(\tau), \mathbf{u}(\tau))\, d\tau, \end{aligned} \tag{7}$$

is subject to the state dynamics

$$\dot{\mathbf{x}}(\tau) = \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)), \tag{8}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the control input, $\mathbf{C}$ is the one step cost integrated over time, and $\mathbf{\Phi}$ is the terminal cost at end time. We impose the boundary constraints

$$\mathbf{e}(\mathbf{x}(\tau_0), \mathbf{x}(\tau_f), \tau_0, \tau_f) = 0 \tag{9}$$

for the initial and terminal states, as well as state and input constraints,

$$\mathbf{h}((\mathbf{x}(\tau), \mathbf{u}(\tau)) \leq 0 \tag{10}$$

in the form of inequalities. It is assumed that $\mathbf{f}$, $\mathbf{\Phi}$, $\mathbf{C}$, $\mathbf{e}$, and $\mathbf{h}$ are nonlinear and smooth functions with respect to $\mathbf{x}$ and $\mathbf{u}$.

C. Pseudospectral discretization

Let N denote the number of collocation points defined for the closed interval $\tau \in [-1, 1]$. The Legendre-Gauss-Lobatto (LGL) points are obtained from the roots of a N^{th}-order Legendre polynomial, $\dot{P}_{N-1}(\tau)$, together with two end points at -1 and 1. As this group of polynomials are orthogonal when evaluated by the L^2 inner product, their derivatives can be expressed in terms of the polynomials themselves, resulting in a convenient and accurate approximation of the differential equations that make up the dynamics.

Let $\mathcal{L}_i, i = 1, \ldots, N$ be the Lagrange basis associated with the collocation points, where

$$\mathcal{L}_i(\tau) = \prod_{j=1}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j}, \quad 1 \leq i \leq N \tag{11}$$

is the Lagrange polynomial of order N. The integral in the objective function can therefore be written as

$$\begin{aligned}\int_{-1}^{1} \mathbf{C}(\mathbf{x}(\tau), \mathbf{u}(\tau))\, d\tau &\approx \sum_{i=1}^{N} \mathbf{C}(\mathbf{x}(\tau_i), \mathbf{u}(\tau_i))\mathbf{w}_i \quad (12)\\ \mathbf{w}_i &= \int_{-1}^{1} \mathcal{L}_i(\tau)\, d\tau, \quad (13)\end{aligned}$$

Similarly, the state trajectory is approximated by the vector-valued polynomial,

$$\mathbf{x}^N(\tau) = \sum_{i=1}^{N} \mathbf{x}_i \mathcal{L}_i(\tau) \quad (14)$$

Differentiating the series and evaluating at the LGL points, $\tau_k, k = 1, \ldots, N$, gives

$$\dot{\mathbf{x}}^N(\tau_k) = \sum_{i=1}^{N} \mathbf{x}_i \dot{\mathcal{L}}_i(\tau_k) = \sum_{i=1}^{N} \mathbf{D}_{ki}\mathbf{x}_i, \quad (15)$$

where $\mathbf{D}_{ki} = \dot{\mathcal{L}}_i(\tau_k)$ is a square matrix called the Lobatto pseudospectral differentiation matrix [7]. Let $\mathbf{X}^{LGL}$ and $\mathbf{U}^{LGL}$ be the state and input approximations at the LGL points. We can write the original optimal control problem, (7)-(10), in the form

$$\min \quad \mathbf{J}(\mathbf{X}_N), \quad (16)$$

subject to

$$\mathbf{D}\mathbf{X}^{LGL} = \mathbf{F}(\mathbf{X}^{LGL}, \mathbf{U}^{LGL}), \quad (17)$$

with the initial condition

$$\mathbf{X}_1 = \mathbf{x}_0. \quad (18)$$

This discrete nonlinear programming problem can be solved by an optimization solver such as SNOPT [20].

V. Results

A. Experimental setup

To optimize the joint motions of the system during a falling trajectory, we define the objective function to minimize the impulse at impact while accounting for the control effort during the trajectory. This is expressed using a terminal constraint $\mathbf{\Phi}(\mathbf{x}(\tau_0), \mathbf{x}(\tau_f), \tau_0, \tau_f)$ in (7). We also include an additional term on the derivative of the control input to penalizing non-smooth and oscillatory control torques.

$$\mathbf{J} = k_1 I + \int_0^T \left(\mathbf{u}^T\mathbf{W}\mathbf{u} + \dot{\mathbf{u}}^T\mathbf{V}\dot{\mathbf{u}}\right)\, dt \quad (19)$$

where k_1 is a scalar weight and $\mathbf{W}$ and $\mathbf{V}$ are positive-definite weight matrices. The dynamic model is taken in the form of (1) and incorporate joint and actuator limits in the form of (10). The values are given in Table II.

TABLE II
State and Input Constraints

	q_1	q_2	q_3	$\dot{q}_1$	$\dot{q}_2$	$\dot{q}_3$	u_1, u_2, u_3
min	-90	-150	0	-800	-800	-800	-1000
max	90	150	180	800	800	800	1000

Two simulated experiments are designed to determine the optimal strategy in both forward and backward directions, compared to the uncontrolled case. In each scenario, we examine the joint trajectories for the three links (q_1, q_2, and q_3). The initial joint angles are set such that the robots are standing up. Small, non-zero values are used to avoid the kinematic singularity that occurs when the links are aligned. Visualization using a simulated robot accompany the joint trajectories for each scenario. An animated video can be found at [21].

B. Uncontrolled falling

Fig. 4 shows the joint trajectories for an uncontrolled fall and and Fig. 3 gives a visualization of the initial and final joint configurations using a simulated robot. Without compensating for the fall, the robot lands on its knees and the upper body falls forward.

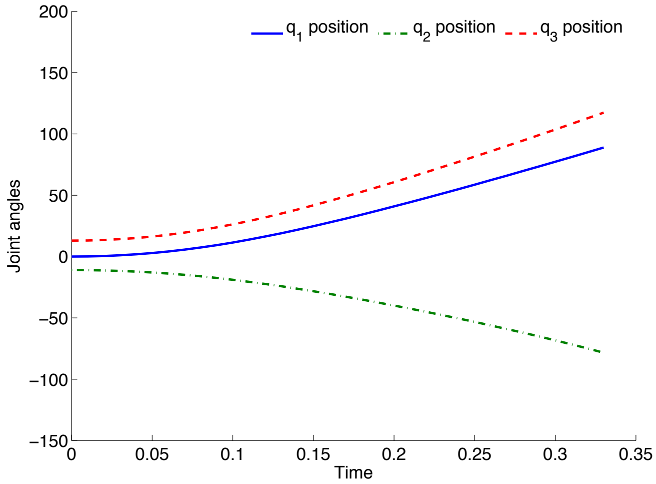


Fig. 4. Uncontrolled falling - joint trajectories move uncontrolled in the direction of the initial condition until impact. The robot lands on its knees and the upper body falls forward.

C. Controlled falling - forward direction

Fig. 5 shows the joint trajectories for a controlled fall. The produced optimized strategy compensates by falling forward in the positive q_1 direction, landing at the knees. The thigh bends backwards until the joint limit at 150° with respect to the shank while simultaneously bends the torso forward. At the final configuration, the robot kneels while keeping the upper body upright. Fig. 3 gives a visualization of the initial and final joint configurations.

D. Controlled falling - backward direction

Fig. 6 shows the joint trajectories for a second falling strategy as the robot falls backwards. In this case, the system falls backward in the negative q_1 direction and lands at the hip joint. While falling, the torso bends forward until reaching the joint limit of 180° with respect to the thigh joint. At the final configuration, the is torso almost upright. Fig. 1 gives a visualization of the initial and final joint configurations.

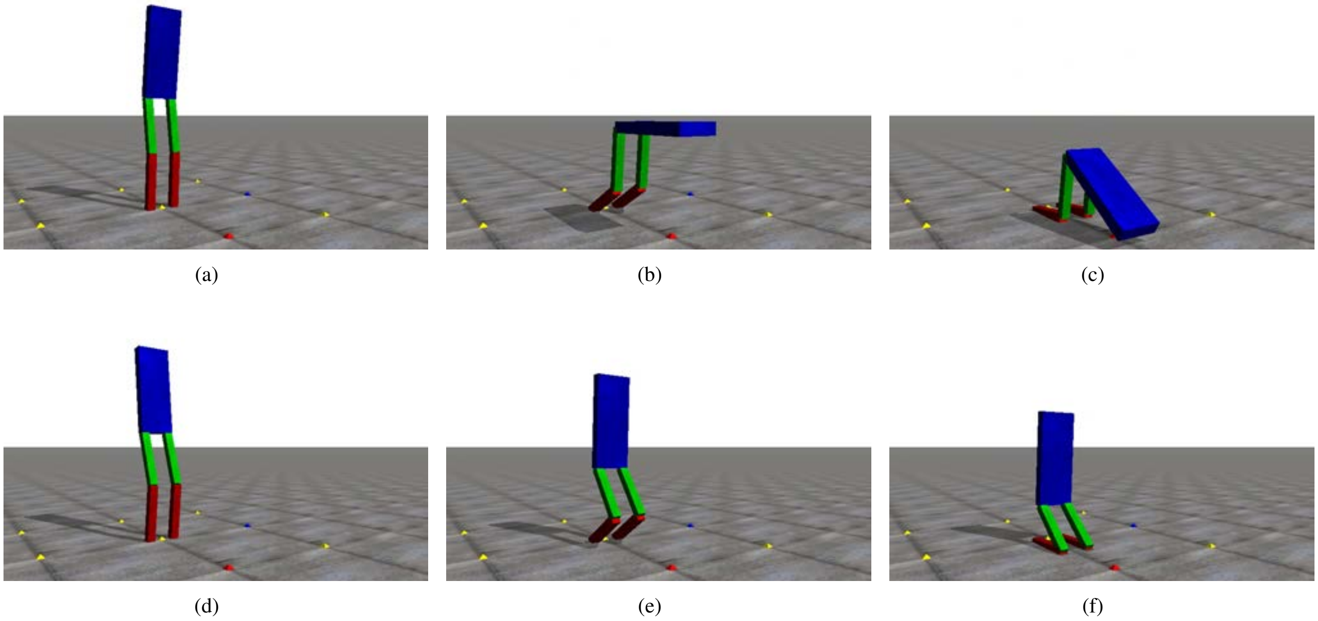


Fig. 3. Top (a)-(c): Uncontrolled fall for a three-link humanoid model where links move uncontrolled in the direction of the initial condition until impact. Bottom (c)-(d): controlled fall in the forward direction, landing at the knees. The thigh bends backwards until hitting a joint limit with respect to the shank while simultaneously bends the torso forward. At the final configuration, the robot kneels while keeping the upper body upright.

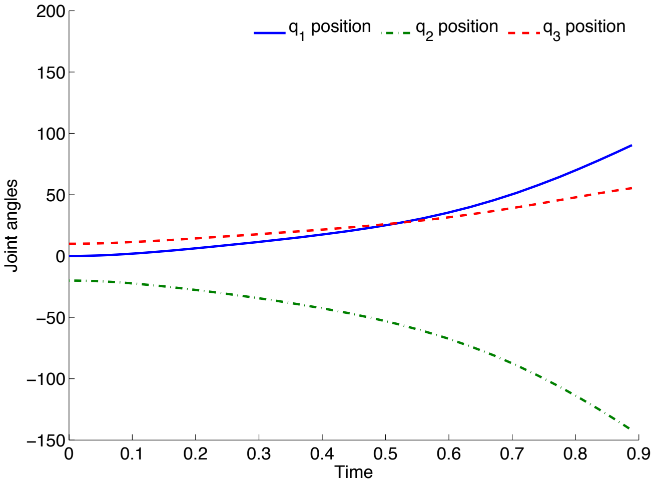


Fig. 5. Controlled falling in the forward direction. The system falls forward in the positive q_1 direction, landing at the knees. The thigh bends backwards until the joint limit at $-150°$ with respect to the shank while simultaneously bends the torso forward. At the final configuration, the robot kneels while keeping the upper body upright. Fig. 3 gives a visualization of the initial and final joint configurations using these trajectories.

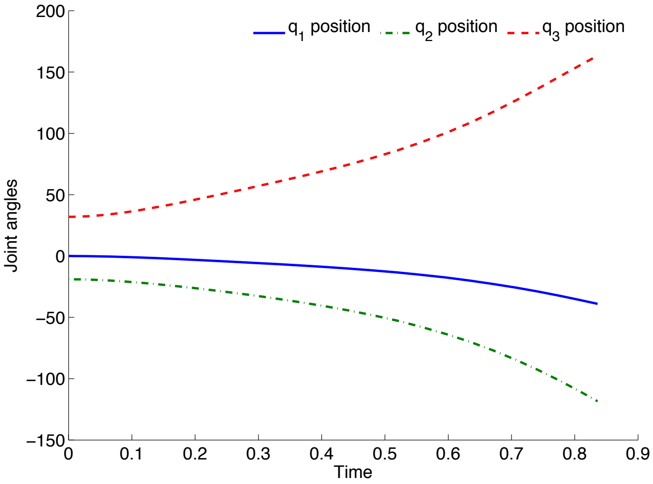


Fig. 6. Controlled falling in the backward direction. The system falls backward in the negative q_1 direction and lands at the hip joint. While falling, the torso bends forward until reaching the joint limit of $180°$ with respect to the thigh joint. At the final configuration, the torso is almost upright. Fig. 1 gives a visualization of the initial and final joint configurations using these trajectories.

E. Discussion

Using the two basic scenarios defined above, we experimented with various combinations of initial conditions and cost parameters. The results show that the impact reduction strategies that emerges from the optimization process can reduce the impulse at impact approximately 40%-70% compared to uncontrolled falling. In reality, these figures are subject to the accuracy of the impact model, the dynamics of the robot, as well as other unmodeled effects such as friction and slippage. For robots with stiff joints due to heavily geared motors, it is easy to modify the current model to include motor inertias and reflected inertias in (1). The control-affine form (2) and the subsequent derivations can be used without modification.

Where in typical collocation-based algorithms, the initial condition plays an important role in converging to an optimum, the scenarios tested here converge given almost any initial trajectory. While we do incorporate input constraints in our formulation, the feasible trajectories generated are far from the boundaries of constraints indicating that the optimal falling strategies do not involve pushing against the actuator limits. For robots of similar configurations and power, we expect that actuators are adequate for maintaining the final configuration of the fall.

We note that while the approach presented is only guaranteed to produce locally optimal solutions, the results generated are consistent with the optimal results presented in [18], which suggest that the "knee landing" strategy for the forward case is likely to be a global optimum for legged systems of this configuration.

All experimental results are generated on a 32-bit system using a Core 2 Duo 2.4GHz machine with 4GB of RAM. For all cases, the results can be generated in under 15 seconds using 40 collocation points. The number of points was determined through several experiments, in which increasing the number of collocation points beyond this value produced little increase in accuracy. While it still cannot be implemented online in a closed-loop sitting, this approach can be used as a basis for trajectory libraries [22], where a group of trajectories are computed offline and interpolated as needed. With a uniform grid sampling the possible configurations before the fall, we expect that a library of trajectories will be easy to compute offline and executed online.

VI. Conclusions

We have presented an optimal control formulation for the humanoid falling problem where we aim to minimize the damage inflicted on the robot during impact. By transcribing the optimal control problem into an NLP, the solution is obtained through numerical optimization. Compared to previous work in this domain, our approach gives a consistent formulation, where the impact dynamics and physical constraints are incorporated naturally into the numerical optimization framework, making it extendable to different models with varying complexity.

Immediate future research will focus on implementing the proposed optimization in a trajectory library setting as well as experimenting with more complex dynamic models, using multiple links for the upper body including head and arms. The proposed optimization framework can be extended to protect specific parts of the body such as computers, sensors, and weakest structural components by assigning higher penalty to important components. Developing a more accurate impact model can significantly improve the performance of the falling strategy as well.

Acknowledgments

This work is supported in part by the National Science Foundation (ECCS-0824077, IIS-0964581, IIS-1017076) and the DARPA M3 program. Jiuguang Wang and Eric Whitman are also supported by the NSF Graduate Research Fellowship Program (GRFP), Grant #0750271. The authors are grateful to Chris Atkeson, Supreeth Achar, Pranav Bhounsule, David Matten, and Varun Ramakrishna for their valuable comments and suggestions.

References

[1] M. Xie, *Fundamentals of Robotics: Linking Perception to Action*, ser. Machine Perception and Artifical Intelligence. World Scientific Publishing Company, 2003.

[2] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *IEEE-RAS International Conference on Humanoid Robots*, Dec. 2006, pp. 200–207.

[3] B. J. Stephens and C. G. Atkeson, "Dynamic Balance Force Control for Compliant Humanoid Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 1248–1255.

[4] K. Yamane and J. Hodgins, "Simultaneous Tracking and Balancing of Humanoid Robots for Imitating Human Motion Capture Data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 2510–2517.

[5] A. Murai and K. Yamane, "A Neuromuscular Locomotion Controller That Realizes Human-Like Responses to Unexpected Disturbances," in *IEEE International Conference on Robotics and Automation*, May 2011, pp. 1997–2002.

[6] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2nd ed., ser. Advances in Design and Control. Society for Industrial and Applied Mathematics, 2009.

[7] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A Unified Framework for the Numerical Solution of Optimal Control Problems Using Pseudospectral Methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.

[8] R. Renner and S. Behnke, "Instability Detection and Fall Avoidance for a Humanoid Using Attitude Sensors and Reflexes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2006, pp. 2967–2973.

[9] J. G. D. Karssen and M. Wisse, "Fall Detection in Walking Robots by Multi-Way Principal Component Analysis," *Robotica*, vol. 27, no. 02, pp. 249–257, 2009.

[10] S. Kalyanakrishnan and A. Goswami, "Learning to Predict Humanoid Fall," *International Journal of Humanoid Robotics*, vol. 8, no. 2, pp. 245–273, Jun. 2011.

[11] U. Nagarajan and A. Goswami, "Generalized Direction Changing Fall Control of Humanoid Robots Among Multiple Objects ," in *IEEE International Conference on Robotics and Automation*, May 2010, pp. 3316–3322.

[12] S.-K. Yun, A. Goswami, and Y. Sakagami, "Safe Fall: Humanoid Robot Fall Direction Change Through Intelligent Stepping and Inertia Shaping," in *IEEE International Conference on Robotics and Automation*, May 2009, pp. 781–787.

[13] 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," *Robotics and Autonomous Systems*, vol. 57, no. 8, pp. 796–807, 2009.

[14] K. Fujiwara, F. Kanehiro, S. Kajita, K. Kaneko, K. Yokoi, and H. Hirukawa, "UKEMI: Falling Motion Control to Minimize Damage to Biped Humanoid Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2002, pp. 2521–2526.

[15] K. Fujiwara, F. Kanehiro, S. Kajita, and H. Hirukawa, "Safe Knee Landing of a Human-Size Humanoid Robot While Falling Forward," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2004, pp. 503–508.

[16] S. Lee and A. Goswami, "Fall on Backpack: Damage Minimizing Humanoid Fall on Targeted Body Segment Using Momentum Control," in *International Conference on Multibody Systems, Nonlinear Dynamics, and Control*, Aug. 2011.

[17] K. Ogata, K. Terada, and Y. Kuniyoshi, "Real-Time Selection and Generation of Fall Damage Reduction Actions for Humanoid Robots," in *IEEE-RAS International Conference on Humanoid Robots*, Dec. 2008, pp. 233–238.

[18] 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," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2007, pp. 456–462.

[19] G. Elnagar, M. Kazemi, and M. Razzaghi, "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, vol. 40, no. 10, pp. 1793–1796, Oct. 1995.

[20] P. Gill, W. Murray, and M. Saunders, "SNOPT: an SQP Algorithm for Large-Scale Constrained Optimization," *SIAM journal on optimization*, vol. 12, no. 4, pp. 979–1006, 2002.

[21] J. Wang, E. C. Whitman, and M. Stilman, http://jw.nebulis.org/publications/.

[22] M. Stolle and C. G. Atkeson, "Finding and Transferring Policies Using Stored Behaviors," *Autonomous Robots*, vol. 29, pp. 169–200, 2010.