Robotica (2013) volume 31, pp. 311–322. © Cambridge University Press 2012
doi:10.1017/S0263574712000203

Biped walking control using a trajectory library

Chenggang Liu†*, Christopher G. Atkeson‡ and Jianbo Su†

†*Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China*
‡*Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA cga@cmu.edu*

(Accepted April 13, 2012. First published online: May 25, 2012)

SUMMARY

This paper presents biped walking control using a library of optimal trajectories. Biped walking control is formulated as an optimal control problem. We use a parametric trajectory optimization method to find the periodic steady-state walking trajectory. As a second stage, we use Differential Dynamic Programming to generate a library of optimal trajectories and locally linear models of the optimal control law, which are used to construct a more global control law. The proposed controller is compared with a trajectory tracking controller using optimal gains. The utility and performance of the proposed method are evaluated using simulated walking control of a planar five-link biped robot.

KEYWORDS: Trajectory optimization; Trajectory library; Robot control.

1. Introduction

The control of biped walking remains a difficult problem because of high dimensionality, nonlinearity, intermittent contact between the feet and the ground, and constraints on kinematics and dynamics, such as joint limitations, the foot clearance requirement, and foot–ground contact conditions.[1] The most common approach to biped walking control is through tracking pre-computed reference trajectories. These trajectories can be computed based on the concept of Zero Moment Point (ZMP),[1] where emphasis is placed on enlarging the stability margin during gait planning.[2,3] Trajectories can also be computed based on the Linear Inverted Pendulum Model (LIPM).[4] By simplifying the dynamics to a linear system, modern control techniques can be used to plan gaits.[5–7] Trajectory optimization of various cost criteria can be used to generate reference trajectories.[8–13] Feedback control methods, such as proportional–integral–derivative (PID) controllers, computed torque, and ZMP feedback control, are used to track reference trajectories.[2,14–16] However, the stability obtained this way may be limited and even small perturbations may lead to a fall.[17] To improve the stability that can be obtained from such a tracking control law, online adaptation of the choice of the reference trajectory was proposed.[17] But the stiff tracking control laws it uses require large joint torques if a deviation from the reference trajectory occurs.[17]

In this paper, we formulate biped walking control as an optimal control problem. Dynamic Programming (DP) can be used to find an optimal control law for constrained nonlinear systems.[18] Using appropriate state space decomposition and/or model reduction, DP has been applied to walking.[19–21] However, when the dimension of the state space is high, the computation and even the storage of the optimal control law becomes difficult.[22] Differential Dynamic Programming (DDP) is a local version of dynamic programming,[22–24] as it applies the principle of optimality in the neighborhood of a candidate trajectory. This allows the coefficients of a quadratic expansion of the value function and a linear expansion of the optimal control law to be computed along the trajectory. These coefficients may then be used to compute an improved trajectory.[22–24] After DDP's convergence to an optimal trajectory, locally linear models of the optimal control law are available. DDP has been applied to biped walking control.[25,26] Morimoto and Atkeson[25] proposed minimax DDP to generate a robust local controller for biped walking. The stability obtained using only one trajectory is limited, and this paper explores using multiple trajectories to achieve robustness. Tassa *et al.*[27] proposed receding horizon DDP to generate time-invariant local controllers. In a more recent paper[28] the authors combine receding horizon control with a stored limit cycle. They describe a similar approach to finding the limit cycle.[29] This work and Liu and Su[30] stress the value of combining online trajectory optimization with stored trajectories, which we will explore for biped walking in future research.

For DDP's convergence to an optimum, a good initial trajectory is important for highly nonlinear and constrained systems, such as bipedal walking. In the present paper, we take advantage of a parametric trajectory optimization method to find coarse initial trajectories for DDP to refine further. Parametric trajectory optimization methods have risen to prominence in solving optimal control problems. In a parametric trajectory optimization method, the state and control variables or the coefficients of their function approximations are taken as optimization variables.[8,9,31–35] An optimal control problem is transcribed into a nonlinear programming problem of finite dimension, which can be solved by general nonlinear programming methods, such as Sequential Quadratic Programming (SQP).[31,36] Parametric trajectory optimization methods are good at handling constraints and are robust in terms of finding a solution when little is known about the final result.[31–33] But parametric trajectory optimization methods take the system dynamics as constraints rather than generating trajectories by integrating the system dynamics, thus they may produce less accurate and sometimes physically incorrect solutions.[31] To improve

* Corresponding author. E-mail: cgliu2008@gmail.com

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press

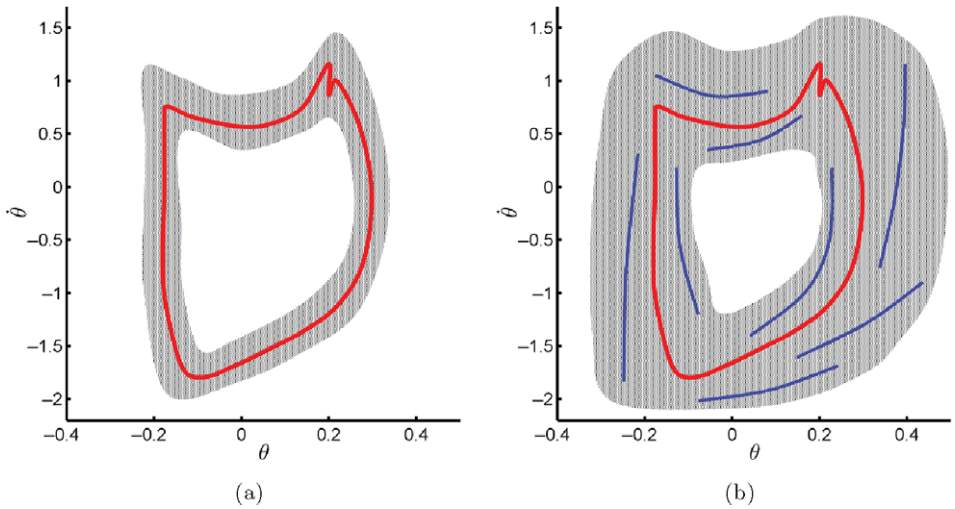


Fig. 1. (Colour online) Sketch of trajectory library control of biped walking. The shaded area indicates where the controller works. (a) The periodic steady-state trajectory. (b) After adding optimal trajectory segments.

the low accuracy of a parametric trajectory optimization method, Hardt[9] combined direct minimization, shooting, and a multiple shooting method. Stryk and Bulirsch[37] proposed a hybrid approach combining a direct collocation method and a multiple shooting method.

Figure 1 sketches the vision of how the proposed method would work for biped walking control. On the left is a plot of the periodic steady-state walking trajectory (a periodic trajectory with the lowest cost) at a specified speed. We use a parametric trajectory optimization method to find the periodic steady-state trajectory. It is then used as an initial trajectory for DDP to further refine and generate locally quadratic models of the value function and locally linear models of the optimal control law around it. The periodic steady-state trajectory can then be stabilized using nearest neighbor control and the resulting basin of attraction might look something like the shaded area in Fig. 1(a).

We use a trajectory library and locally linear controllers to construct a walking controller. As a second step, DDP can quickly generate many optimal trajectory segments starting from nearby states. The resulting basin of attraction of the trajectory library might look something like the shaded area in Fig. 1(b). In contrast to most trajectory optimization approaches to walking, we formulate the trajectory optimization with an infinite time horizon so that the corresponding value function and control law are time-independent and functions only of state.

A number of efforts have been made to use collections of trajectories to represent policies. Early work in this area is reviewed by Atkeson and Stephens[38] and Stolle and Atkeson.[39] Additional works include Breitner[40] and Murray.[41] There has been a great deal of work in computer animation and video games using trajectory libraries, for example Safonova and Hodgins.[42] Authors in refs. [17, 43, 44] used trajectory libraries for controlling humanoid walking using manually designed and optimized trajectories. A related technique in path planning is the creation of Probabilistic Roadmaps (PRMs),[45] which is a form of trajectory library. Atkeson and Morimoto[26,46] used locally linear controllers along optimal trajectories to construct a nonparametric representation of a global control law, and information transfer between stored trajectories to form sets of globally optimized trajectories for control. Tassa *et al.*[27] used a trajectory library to synthesize a global controller for a simulated multi-link swimming robot. Reist and Tedrake[47,48] used local linear-quadratic regulator (LQR) controllers to construct a nonlinear feedback policy. In our previous work, a trajectory library was applied to standing balance of humanoid robots.[49]

This paper has the following organization. In Section 2 the dynamics of the simulated five-link robot model is described. In Section 3 the optimal control problem of biped walking is formulated. Then a parametric trajectory optimization method is proposed, which solves for the periodic steady-state trajectory. Differential Dynamic Programming with an infinite time horizon is introduced, which generates optimal trajectories and time-invariant locally linear models of the optimal control law. In Section 4 simulation results are presented. Conclusions and future work are discussed in Section 5.

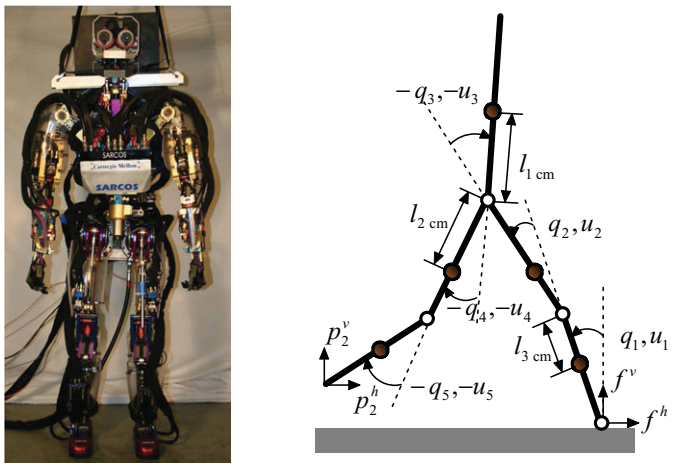
2. Robot Model

The robot, as shown in Fig. 2(b), is assumed to be planar and consist of a torso and two identical legs with knees. Furthermore, all body parts have mass, and are rigid and connected with revolute joints. It is modeled on our Sarcos Primus System hydraulic humanoid robot shown in Fig. 2(a) and its kinematic and dynamic parameters are listed in Table I. The Sarcos robot uses hydraulic actuators and load sensors to control torque at each joint.[50] In our planar model, all walking cycles take place in the sagittal plane and consist of successive phases of single support and impact. There is no double support. The walking cycle has two mathematical models: ordinary differential equations, describing the dynamics during the single-support phase, and an impact model, describing the instantaneous change in the velocities and coordinate relabeling after the impact event.

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press

Table I. Physical parameters of the simulated robot.

	Calf	Thigh	Torso
Mass (kg)	6.90	5.68	50.00
Moment of inertia (kg·m^2)	0.15	0.10	1.50
Length (m)	0.38	0.39	0.80
l_{cm} (m)	0.24	0.19	0.29



(a) (b)

Fig. 2. (Colour online) (a) Sarcos Primus System hydraulic humanoid robot. (b) Simplified structure of the biped robot used for our study.

2.1. Single-support model

In the single-support phase, the dynamic model of the robot has 5 degrees of freedom (DoF). Let $\mathbf{q} := (q_1, \ldots, q_5)^{\mathrm{T}}$ be the generalized coordinates describing the configuration of the robot depicted in Fig. 2(b). The same dynamic model is used no matter which leg is the stance leg, and the coordinates are relabeled after the impact event. The dynamics equations can be derived using the method of Lagrange. The result is a standard second-order system,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}(\mathbf{q})\mathbf{u}, \tag{1}$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{5\times5}$ is the inertia matrix, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^5$ is the vector of centrifugal, Coriolis, and gravity forces, and $\mathbf{B}(\mathbf{q})$ is the matrix defining how the joint torques $\mathbf{u} := (u_1, \ldots, u_5)^{\mathrm{T}}$ enter the model. Though a stance foot is not explicitly modeled while on the ground, the ankle may apply a torque u_1 between the calf and the ground in the pitch direction. Defining $\mathbf{x} := (\mathbf{q}^{\mathrm{T}}, \dot{\mathbf{q}}^{\mathrm{T}})^{\mathrm{T}}$, the second-order system of Eq. (1) can be written in state space form as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}^{-1}(\mathbf{q})(-\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}(\mathbf{q})\mathbf{u}) \end{bmatrix} = f(\mathbf{x}, \mathbf{u}). \tag{2}$$

2.2. Impact model

The impact between the swing leg and the ground is modeled as a contact between two rigid bodies. We assume the instantaneous contact of the swing leg end with the ground results in no rebound and no slipping of the swing leg. The impulsive forces due to the impact result in an instantaneous change in the velocities, but there is no instantaneous change in the positions. The impact model results in a smooth map:[51]

$$\mathbf{x}^+ = H(\mathbf{x}^-), \quad \mathbf{x}^- \in \mathcal{S}, \tag{3}$$

where $\mathbf{x}^-$ is the value of the state just prior to impact, $\mathbf{x}^+$ is the value of the state just after impact, and $\mathcal{S} = \{\mathbf{x} | p_2^v(\mathbf{x}) = 0, \dot{p}_2^v(\mathbf{x}) < 0\}$ on flat ground (the definition of p_2^v is shown in Fig. 2(b)). The function H represents the velocity change caused by impact and the relabeling of the robot's coordinates, which makes the swing leg become the stance leg.

The dynamics of the overall biped robot is described by a nonlinear system with discontinuous dynamics. The robot mechanical system evolves according to ordinary differential Eq. (2) until the state enters set S when impact occurs and a new initial state from which the robot model evolves is given by Eq. (3). The ground reaction forces, f^h and f^v as depicted in Fig. 2(b), can be calculated using the inverse dynamics of the robot.

3. Control Using Trajectories and Local Models

3.1. Optimal control formulation

Biped walking control is formulated as an infinite time optimal control problem. The cost functional should have a discount factor to remain finite,

$$J := \int_0^\infty e^{-\beta t} c(\mathbf{x}(t), \mathbf{u}(t))\, \mathrm{d}t, \tag{4}$$

where $\beta > 0$ is the discount rate and $c(\cdot, \cdot)$ is the cost rate function. A state-feedback control law, $\mathbf{u} = \mathbf{u}^*(\mathbf{x})$, is optimal if the cost functional J is minimized given the system followed $\mathbf{u}^*(\mathbf{x})$ subject to the dynamics Eqs. (2) and (3). In addition, constraints on state variables of the form

$$\mathbf{x}_l \le \mathbf{x}(t) \le \mathbf{x}_u, \tag{5}$$

and control variables of the form

$$\mathbf{u}_l \le \mathbf{u}(t) \le \mathbf{u}_u \tag{6}$$

can be applied.

3.2. Periodic steady-state trajectory optimization

To generate an initial limit cycle, we can take one phase of single support and one phase of impact as one walking cycle. Given the periodic trajectory starts and ends at the dynamics discontinuity, we uniformly divide the time interval $(0, T)$ into N small time intervals, $0 = t_0 < t_1 < \ldots < t_N = T$, and use the Runge–Kutta scheme to discretize the dynamics equations of Eq. (2) during the single-support phase:[31]

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{u}_k), \tag{7}$$

where $\mathbf{x}_k$ and $\mathbf{u}_k$ denote the values of the state and the control at the sampled points, respectively. The cost functional on

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press

a periodic trajectory can then be approximated by the cost function

$$J = \frac{1}{1-\lambda}\left[\lambda^N \phi(\mathbf{x}_N) + \sum_{k=0}^{N-1} \lambda^k L(\mathbf{x}_k, \mathbf{u}_k)\right], \quad (8)$$

where $\lambda = \exp(-\beta T/N)$ is the discount factor, $\phi(\cdot)$ is the impact cost, and $L(\cdot,\cdot)$ is the one-step cost function defined as

$$L(\mathbf{x}, \mathbf{u}) = c(\mathbf{x}, \mathbf{u})T/N. \quad (9)$$

Thus, we can find the periodic steady-state trajectory with minimal cost through trajectory optimization. By treating the period T and the state and control variables at sampled points as optimization variables, the trajectory optimization problem is cast as a parametric nonlinear programming (NLP) problem. It minimizes the cost function (Eq. (8)) subject to the discrete time dynamics Eq. (7), constraints on the state (Eq. (5)), on the control (Eq. (6)), and

$$H(\mathbf{x}_N) = \mathbf{x}_0 \quad (10)$$

$$\mathbf{x}_N \in \mathcal{S}. \quad (11)$$

We solve this NLP problem with SNOPT.[36]

3.3. Optimization criteria and constraints

The one-step cost function $L(\mathbf{x}, \mathbf{u})$ we use penalizes the joint torques, $\mathbf{u}^{\mathrm{T}}\mathbf{W}_u\mathbf{u}$, the walking speed error, $w_v(\dot{p}_3^h(\mathbf{x}) - v_d)^2$, and the horizontal component of the ground reaction force, $w_r(f^h(\mathbf{x}))^2$, where $\mathbf{W}_u$, w_v, and w_r are penalty weights, v_d and $\dot{p}_3^h(\mathbf{x})$ are, respectively, the desired walking speed and the horizontal velocity of the hip, and f^h is the horizontal component of the ground reaction force (as shown in Fig. 2(b)). $L(\mathbf{x}, \mathbf{u})$ also has some "shaping" terms: a penalty on the height of the swing foot when it is below a certain height h_d, $w_c(p_2^v - h_d)^2$, and a penalty on the angular velocity of the torso, $w_t(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)^2$. The impact cost $\phi(\mathbf{x})$ penalizes the impulse $\mathbf{F}$ during impact as $\mathbf{F}^{\mathrm{T}}\mathbf{W}_I\mathbf{F}$, where $\mathbf{W}_I$ is a weight matrix.

The robot's mechanism has some limitations on the ranges of joint angles and joint velocities, which are taken as constraints on the state (Eq. (5)), and on the ranges of joint torques, which are taken as constraints on the control (Eq. (6)). The physics of contact between the stance foot and the ground also introduces some constraints. Although we do not model the foot, the horizontal position of the center of pressure (CoP) with respect to the end of stance leg can be calculated by $x_{cop} = u_1/f^v$, where u_1 is the ankle torque of the stance leg and f^v is the vertical component of the ground reaction force (as shown in Fig. 2(b)). To keep the stance foot flat on the ground, the CoP must be kept inside the foot-support region, $x_{cop} \in \Omega$.[1] To avoid slipping, the ground reaction force must be kept inside the friction cone, $|f^h/f^v| \leq \mu$, where μ is the static friction coefficient. We penalize the magnitudes of the ankle torque, u_1, and the horizontal component of the ground reaction force, f^h, so that the result of trajectory optimization can satisfy these foot-ground contact constraints.

3.4. Nearest neighbor control using local models

We use DDP to further refine the periodic trajectory given by Section 3.2, generate additional trajectory segments, and generate local models of the value function and the optimal control law at the same time. Discrete time DDP takes advantage of the Bellman equation,[22–24] which is

$$V(\mathbf{x}_k) = \min_{\mathbf{u}_k}[L(\mathbf{x}_k, \mathbf{u}_k) + \lambda V(F(\mathbf{x}_k, \mathbf{u}_k))] \quad (12)$$

during the single-support phase or

$$V(\mathbf{x}_k) = \phi(\mathbf{x}_k) + V(H(\mathbf{x}_k)) \quad (13)$$

at impact. $V(\mathbf{x}_k)$ describes the minimal cost-to-go when the optimal feedback control, $\mathbf{u} = \mathbf{u}^*(\mathbf{x})$, is applied. By formulating trajectory optimization with an infinite time horizon, $V(\mathbf{x}_k)$ becomes a time-invariant function only of state.

Given the jth candidate trajectory $\mathbf{x}^j$, DDP integrates the first and second partial derivatives of the value function backward to compute an improved value function and control law. The update rule needs to accommodate the discount factor and is given by[26]

$$Q = L(\mathbf{x}, \mathbf{u}) + \lambda V(F(\mathbf{x}, \mathbf{u})), \quad (14)$$

$$Q_x = L_x + \lambda V_x F_x, \quad (15)$$

$$Q_u = L_u + \lambda V_x F_u, \quad (16)$$

$$Q_{xx} = L_{xx} + \lambda F_x^{\mathrm{T}} V_{xx} F_x + \lambda V_x F_{xx}, \quad (17)$$

$$Q_{xu} = L_{xu} + \lambda F_x^{\mathrm{T}} V_{xx} F_u + \lambda V_x F_{xu}, \quad (18)$$

$$Q_{uu} = L_{uu} + \lambda F_u^{\mathrm{T}} V_{xx} F_u + \lambda V_x F_{uu}, \quad (19)$$

$$\mathbf{K} = Q_{uu}^{-1} Q_{ux}, \quad (20)$$

$$\Delta\mathbf{u} = Q_{uu}^{-1} Q_u, \quad (21)$$

$$V_x(k-1) = Q_x - Q_u\mathbf{K}, \quad (22)$$

$$V_{xx}(k-1) = Q_{xx} - Q_{xu}\mathbf{K}, \quad (23)$$

where subscripts x and u indicate partial derivatives. These partial derivatives are computed as a backward sequence starting from the end of the trajectory. For a periodic steady-state trajectory, we use a second-order Taylor series approximation of the value function at the beginning of the trajectory, $V(0)$, and a linear approximation of H to approximate the value function at the end of the trajectory, $V(N)$:

$$\begin{aligned} V(N) &= \phi(\mathbf{x}_N) + V(H(\mathbf{x}_N)) \\ &\approx \phi(\mathbf{x}_N) + V(\mathbf{x}_0) + V_x(0)(H(\mathbf{x}_N) - \mathbf{x}_0) \\ &\quad + \frac{1}{2}(H(\mathbf{x}_N) - \mathbf{x}_0)^{\mathrm{T}} V_{xx}(0)(H(\mathbf{x}_N) - \mathbf{x}_0). \end{aligned} \quad (24)$$

Therefore,

$$\begin{aligned} V_x(N) &= \phi_x(\mathbf{x}_N) + V_x(0)H_x(\mathbf{x}_N) \\ &\quad + (H(\mathbf{x}_N) - \mathbf{x}_0)^{\mathrm{T}} V_{xx}(0)H_x(\mathbf{x}_N), \end{aligned} \quad (25)$$

$$V_{xx}(N) = \phi_{xx}(\mathbf{x}_N) + H_x(\mathbf{x}_N)^{\mathrm{T}} V_{xx}(0)H_x(\mathbf{x}_N). \quad (26)$$

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press

A new updated initial state is given by

$$\mathbf{x}_0^{j+1} = \mathbf{x}_0^j - \epsilon V_{xx}(0)^{-1} V_x(0), \tag{27}$$

and a new updated trajectory $\mathbf{x}^{j+1}$ is generated by integrating the system dynamics forward in time using the linear feedback law:

$$\mathbf{u}^{j+1} = \mathbf{u}^j - \mathbf{K}[\mathbf{x}^{j+1} - \mathbf{x}^j] - \epsilon \Delta \mathbf{u}, \tag{28}$$

where $\epsilon \in (0, 1)$. These processes are repeated until convergence to the optimum is obtained. Before integrating the system dynamics forward for the first time, the value function and its partial derivatives are integrated backward until they are nearly unchanged along the initial trajectory given by SNOPT. The amount of backward integration is determined by the discount factor. In practice, the inversion of Q_{uu} must be conditioned. When $\mathbf{Q}_{uu}$ is close to singular, a Levenberg–Marquardt-like scheme can be used.[52] Also, the initial state update (Eq. (27)) and the control sequence update (Eq. (28)) are performed with an adaptive line search of ϵ.[23] To prevent the second-order partial derivatives of the value function from becoming infinite during infinite horizon value iteration, very small quadratic penalties on the deviations from a candidate trajectory are used. This approach is similar to regularizing by adding positive numbers to the diagonal of V_{xx} on each iteration.[53] To satisfy the constraint on knee angle, springs and dampers are used to prevent the knees from overextending during forward simulation. The control variables on the left-hand side of Eq. (28) are limited to satisfy constraints (Eq. (6)). Constraints (Eqs. (10) and (11)) are approximated by augmenting the impact cost ϕ with quadratic penalties on constraint violations (the penalty weight currently used is 10^4).

Byproducts of DDP are locally linear models of the optimal control law, $\mathbf{u} = \mathbf{u}_k^* - \mathbf{K}_k(\mathbf{x} - \mathbf{x}_k^*)$, where $\mathbf{x}_k^*$ and $\mathbf{u}_k^*$ are respectively the sampled state and the corresponding control on the optimal trajectory, and $\mathbf{K}_k$ is given by Eq. (20). By formulating trajectory optimization with an infinite time horizon, these local models are time-invariant and spatially localized.

To construct a control law from spatially localized local models, the simplest way is to use the nearest local model to predict an optimal control for the current state. Defining a Euclidean distance as

$$\Delta \mathbf{x}^{\mathrm{T}} \mathbf{D} \Delta \mathbf{x}, \tag{29}$$

where $\mathbf{D}$ is a distance metric and $\Delta \mathbf{x} := \mathbf{x} - \mathbf{x}_k^*$ is the deviation of current state $\mathbf{x}$ from sampled state $\mathbf{x}_k^*$, nearest neighbor control is given by

$$\mathbf{u} = \mathbf{u}(\mathbf{x}) = \bar{\mathbf{u}} - \bar{\mathbf{K}}(\mathbf{x} - \bar{\mathbf{x}}), \tag{30}$$

where $\bar{\mathbf{x}}$ is the closest $\mathbf{x}_k^*$ to $\mathbf{x}$, and $\bar{\mathbf{u}}$ and $\bar{\mathbf{K}}$ are the corresponding $\mathbf{u}_k^*$ and $\mathbf{K}_k$. The distance metric $\mathbf{D}$ is a diagonal matrix and its elements for joint angles and joint angular velocities are respectively 1 and 10^{-4} according to their ranges. To keep the stance foot flat on the ground, ankle torque u_1 is limited by the constraint on CoP, $u_1/f^v \in \Omega$, before it is applied to the ankle joint. The control variables on the left-hand side of Eq. (30) are also limited to satisfy constraints (Eq. (6)).

Table II. Parameters used by trajectory optimization.

$\mathbf{W}_u$	10^{-3}diag(10, 1, 1, 1, 1)	w_v	10^2
$\mathbf{W}_l$	diag(10^{-1}, 10^{-3})	w_c	10^3
w_r	10^{-6}	h_d	0.05
w_t	10	β	0.5
$\mathbf{x}_u$	(0.8, 0.0, 0.8, 0.8, 0.8, 5.0, 5.0, 5.0, 5.0, 5.0)$^{\mathrm{T}}$	$\mathbf{x}_l$	$-$(0.8, 0.8, 0.8, 0.8, 0.0, 5.0, 5.0, 5.0, 5.0, 5.0)$^{\mathrm{T}}$
$\mathbf{u}_u$	(50.0, 500, 500, 500, 500)$^{\mathrm{T}}$	$\mathbf{u}_l$	$-$(50.0, 500, 500, 500, 500)$^{\mathrm{T}}$

3.5. Trajectory library

Outside the basin-of-attraction of the periodic steady-state trajectory, the control law is expected to perform poorly. One solution to this problem is to fill the portion of state space of interest with a library of non-periodic optimal trajectory segments and consider all of them during the nearest neighbor lookup.[26,27,49,54]

To generate additional optimal trajectory segments, we use the nearest neighbor control and existing trajectories to generate a new fixed-length trajectory segment starting from a nearby state. This trajectory is then re-optimized by DDP. For the update rule at the dynamics discontinuity, ϕ, H, and 1, respectively, substitute for L, F, and λ. The value function at the end point of the trajectory segment, $V(N)$, is approximated by a local model of the value function at the closest state in the library.

To decide where to add a new trajectory, one approach is based on "failure testing." We apply impulsive perturbations of different magnitudes at different locations and time on the simulated robot. If the simulated robot falls down, we add a new optimal trajectory segment to the library. To keep the size of the library small and also satisfy the performance requirement, we can generate a trajectory library on an adaptive grid based on the simulated performance. If the cost's reduction is larger than a given performance bound after re-optimization, a newly generated optimal trajectory segment will be stored in the library.[49]

4. Results

We used 20 time steps for SNOPT to solve for the periodic steady-state trajectory and 100 time steps for DDP to further refine it and generate additional optimal trajectory segments for a walking speed of 0.5 m/s. The parameters used for trajectory optimization are listed in Table II. The stick diagram of the robot's motion with the periodic steady-state trajectory is shown in Fig. 3(a), in which the configuration of the robot is drawn every 50 ms. The torques are shown in Fig. 3(b). The norms of the feedback gain vectors for each control are shown in Fig. 3(c).

We used the steady-state trajectory and its local models to initialize the trajectory library. We applied horizontal impulsive perturbations of different magnitudes at hip during walking. If the simulated robot fell down after perturbation, we added a new trajectory segment to the library. For example, the simulated robot fell down due to a 12 Newton-seconds perturbation using only the

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press

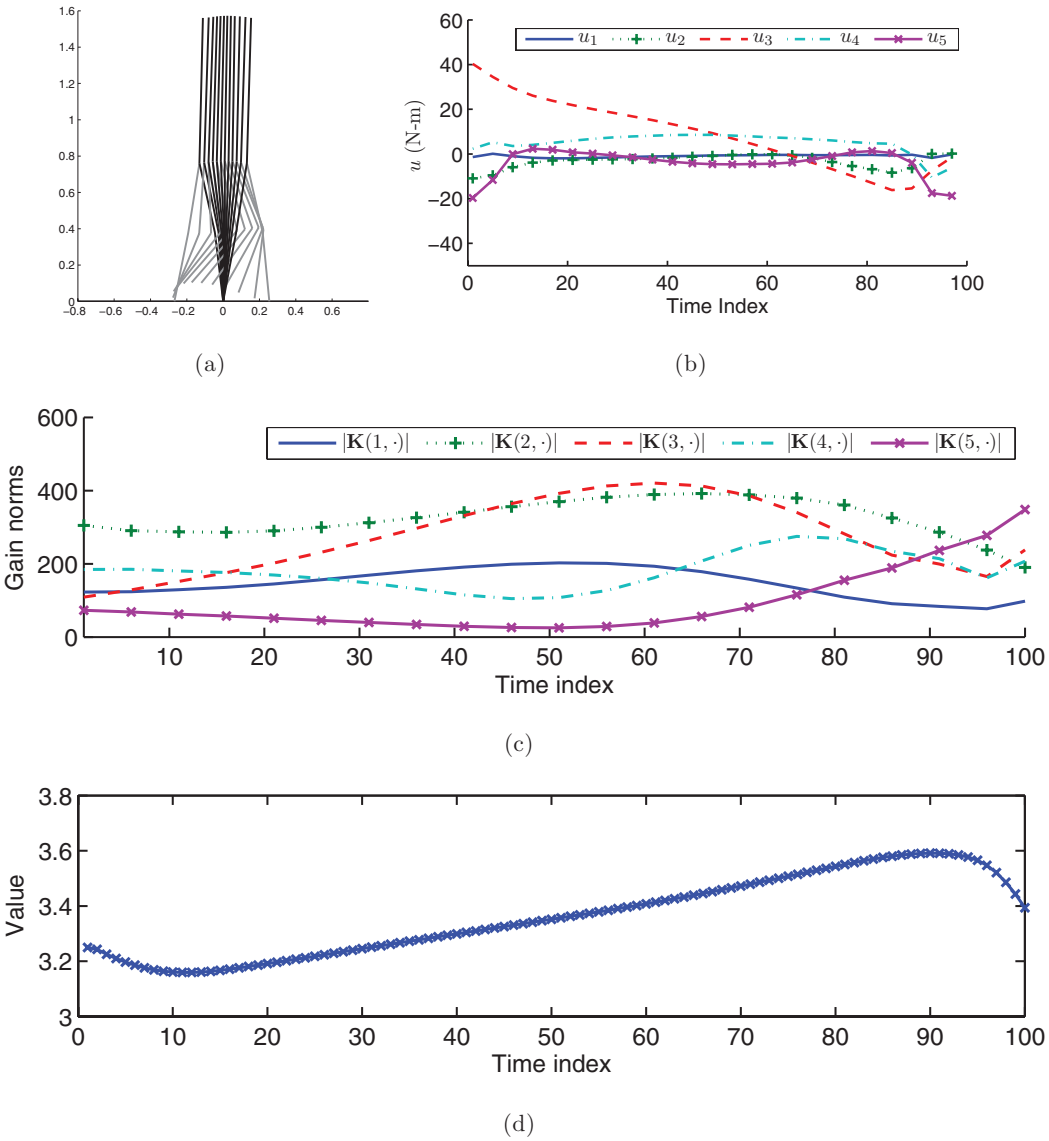
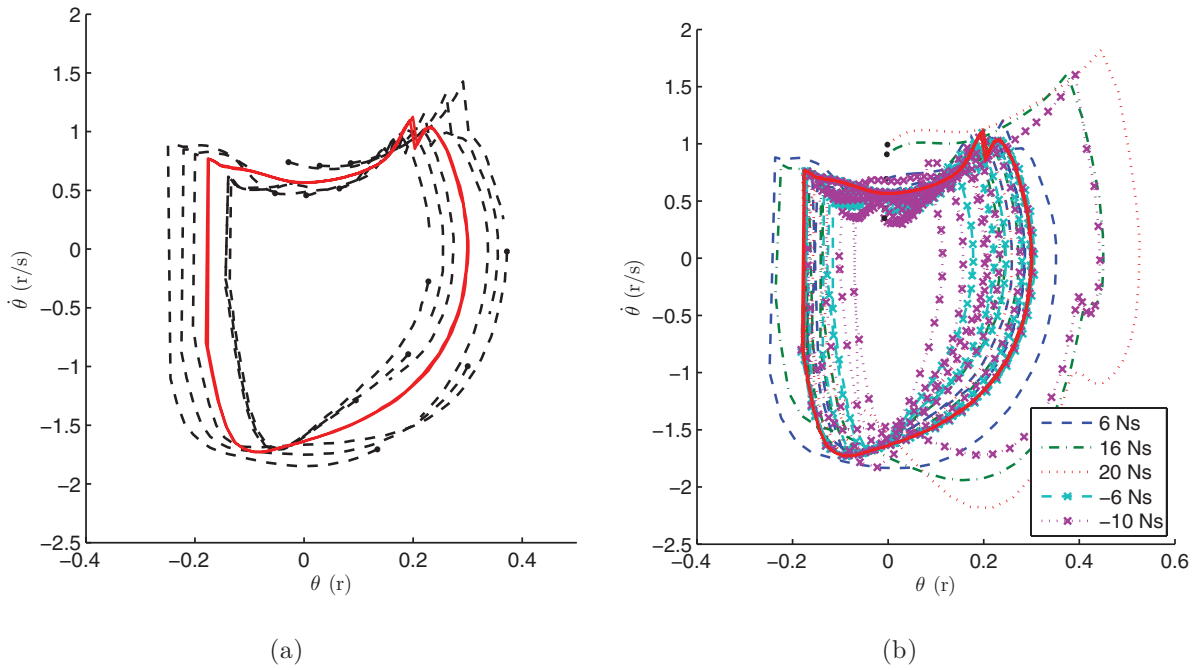


Fig. 3. (Colour online) The periodic steady-state trajectory at 0.5 m/s walking speed. (a) The stick diagram of the robot motion with the periodic steady-state trajectory. The configuration of the robot is drawn every 50 ms. (b) The joint torques on the steady state trajectory. u_1, u_2, and u_3 are those of the ankle, the knee, and the hip pitch of the stance leg, respectively, and u_4 and u_5 are those of the hip pitch and the knee of the swing leg, respectively. (c) The norms of the feedback gain vectors for each control, $|\mathbf{K}(i,\cdot)|$, on the steady-state trajectory, where $u_i = u_i^* - \mathbf{K}(i,\cdot)(\mathbf{x} - \mathbf{x}^*)$. (d) The values of the value function on the steady-state trajectory.

periodic steady-state trajectory. We added an optimal trajectory segment starting from the state just after a 10 Newton-seconds perturbation to the library. Now the proposed controller could handle up to a 20 Newton-seconds perturbation. We use phase portraits of the motions of a virtual leg to show low dimensional projections of optimal trajectories (Fig. 4). The virtual leg is defined as the line connecting the end of one leg to hip. Its pitch angle θ and angular velocity $\dot{\theta}$, with respect to the upright in the world frame, are positive clockwise. An example trajectory library is shown in Fig. 4(a). It contains the periodic steady-state trajectory and 12 optimal trajectory segments starting from the states just after -4 or 10 Newton-seconds perturbations at the hip at different times. The proposed controller using this trajectory library was evaluated using perturbations of -10, -6, 6, 16, and 20 Newton-seconds at the hip. As shown in Fig. 4(b), all trajectories eventually return to the periodic steady-state trajectory. In general, we found that trajectory libraries needed only a relatively small number of entries (for example 12 trajectory segments and a periodic trajectory in this case) to handle reasonably sized perturbations. For the 16 Newton-seconds impulsive perturbation at the hip, the position x_{com} and the velocity $\dot{x}_{\text{com}}$ of the center of mass and the ground reaction forces are shown in Fig. 5(a). It shows that the center of pressure is kept inside the foot-support region, $x_{cop} \in [-0.1, 0.1]$, and the ground reaction force is kept inside the friction cone, $f^h/f^v \in [-1, 1]$. As shown in Fig. 5(b), the trajectory converges to the periodic steady-state trajectory and the walking speed is maintained at 0.5 m/s after the perturbation.

For continuous perturbations, the controller's responses to a continuous forward push of 10 Newtons lasting 2 s are shown in Fig. 6(a). It shows that the robot walks slightly faster than the desired walking speed (0.5 m/s) when the push is applied. The phase portrait of the motion of one virtual leg in Fig. 6(b) shows that the proposed controller drives the robot's

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press



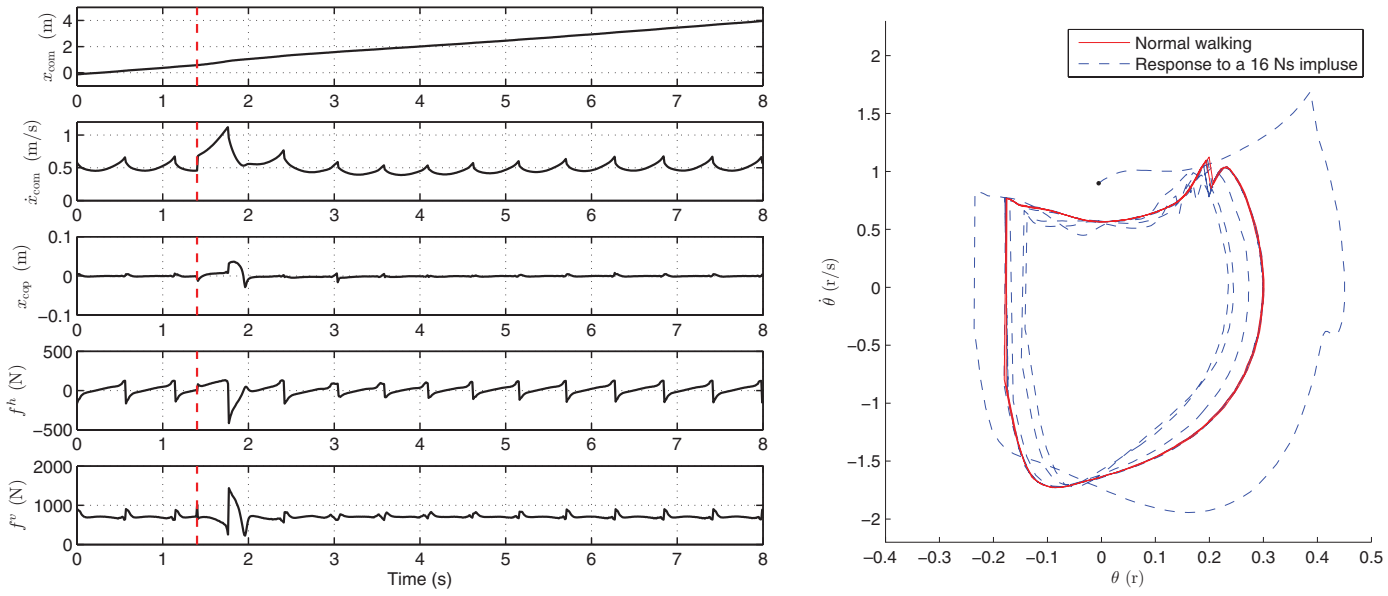
(a) (b)

Fig. 4. (Colour online) The trajectory library and responses to perturbations. The phase portraits of the motions of one virtual leg are shown. The solid line is a periodic steady-state trajectory. Note that this is only a 2D projection of 10D state space, so the trajectories may cross. The discontinuities of the phase portraits are caused by impulsive perturbations. (a) An example trajectory library. The optimal trajectory segments are shown as dashed lines. (b) Responses to novel impulsive perturbations.

trajectory back to the limit cycle of normal walking after the push.

The proposed controller was also evaluated using walking on inclines between -5° and 5°. We only used the periodic steady-state trajectory. Simulation results show the proposed controller generated using a level ground still works. As shown in Fig. 7(a), the average velocity of the center of mass changes only slightly from 0.5 m/s. Fig. 7(b) shows that the controller drives the robot's trajectory to a different limit cycle when it walks on an incline of 5°.

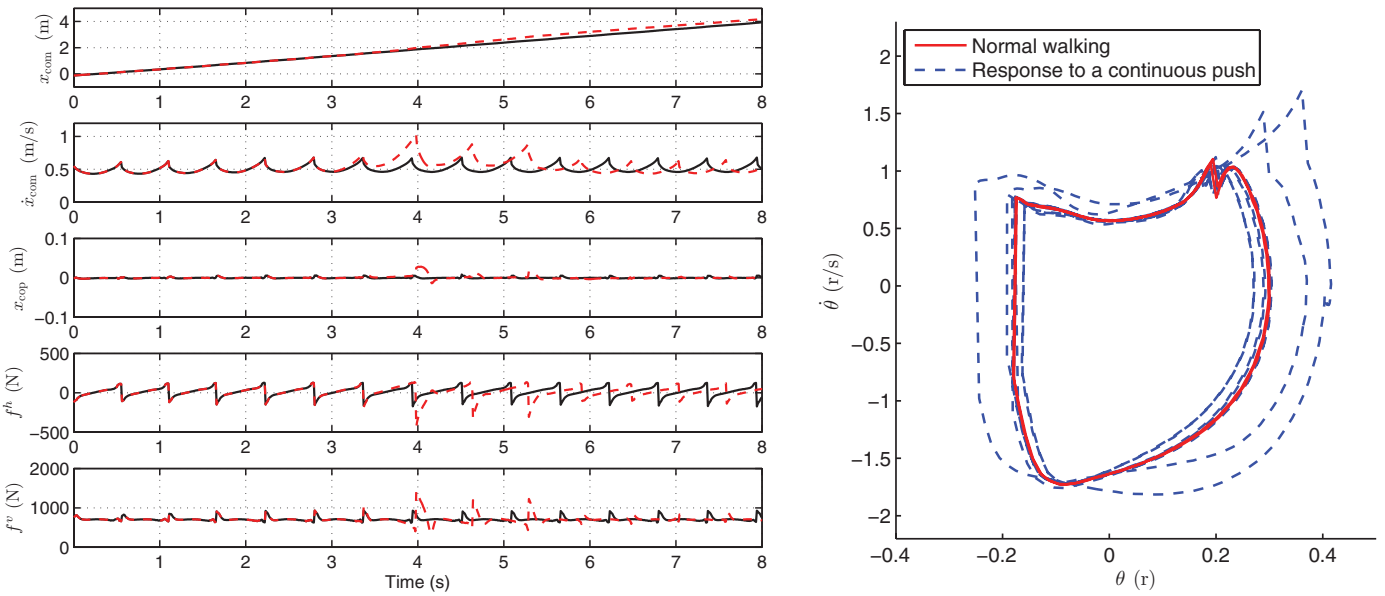
We evaluated the proposed controller using walking control with a model error of $+30\%$ more torso mass. As shown in Fig. 8(a), the vertical component of the ground reaction force increases but the robot can still walk. The walking speed changes very little in this simulation. As shown in Fig. 8(b), the controller drives the robot's



(a) (b)

Fig. 5. (Colour online) Responses to a 16 Newton-seconds perturbation at 1.4 s (the vertical dashed lines in red). (a) The position and the velocity of the center of mass, the position of CoP, and the ground reaction forces. (b) The phase portrait of the motion of one virtual leg. The discontinuity of the phase portrait is caused by impulsive perturbation.

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press



(a) (b)

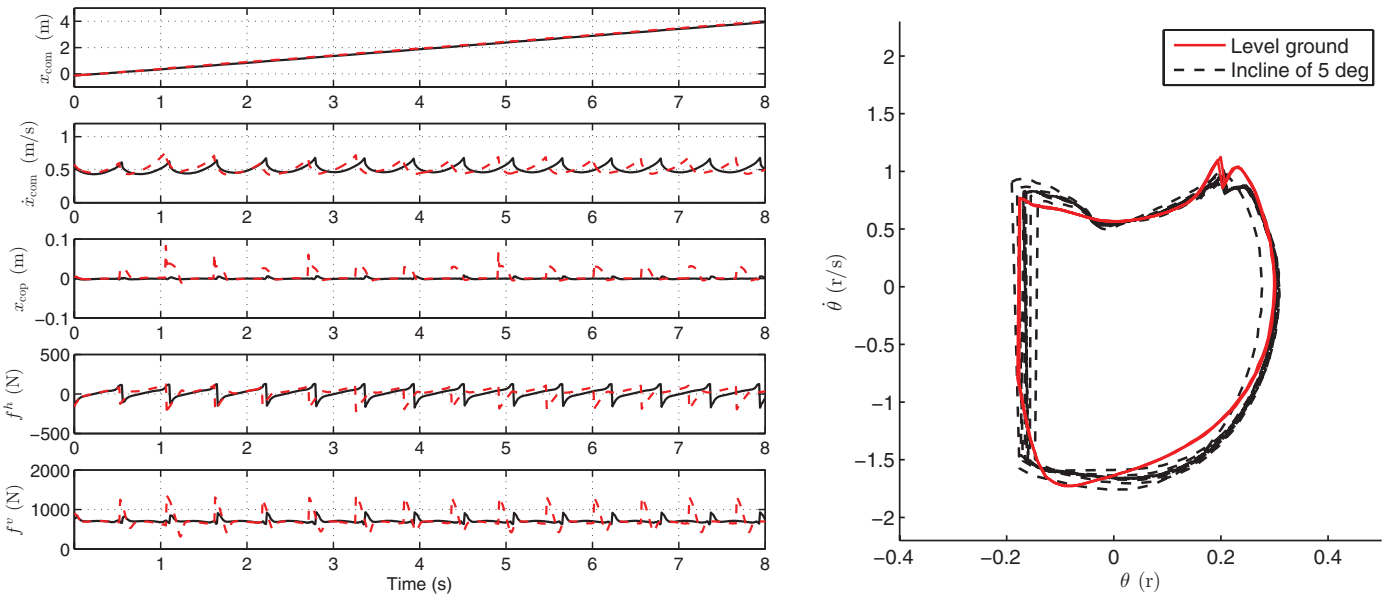
Fig. 6. (Colour online) Responses to a continuous forward push of 10 Newtons from 3 to 5 s as dashed lines. The solid lines are responses of normal walking. (a) The position and the velocity of the center of mass, the position of the CoP, and the ground reaction forces. (b) The phase portrait of the motion of one virtual leg.

trajectory to a different limit cycle because of the model error.

Our approach is able to handle different contact models. An analytical foot–ground contact model (rigid body impact model) is employed for offline controller construction. The resultant controller is tested using other foot–ground contact models, such as a spring-damper model. For the spring-damper contact model, the joint between the stance leg's end and the ground is modeled as a planar joint of 3 DoF and the whole system has 7 DoF. The ground reaction force of each leg's end is calculated by

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix} = -\begin{bmatrix} k_x & 0 & c_x & 0 \\ 0 & k_y & 0 & c_y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \dot{x} \\ \dot{y} \end{bmatrix}, \tag{31}$$

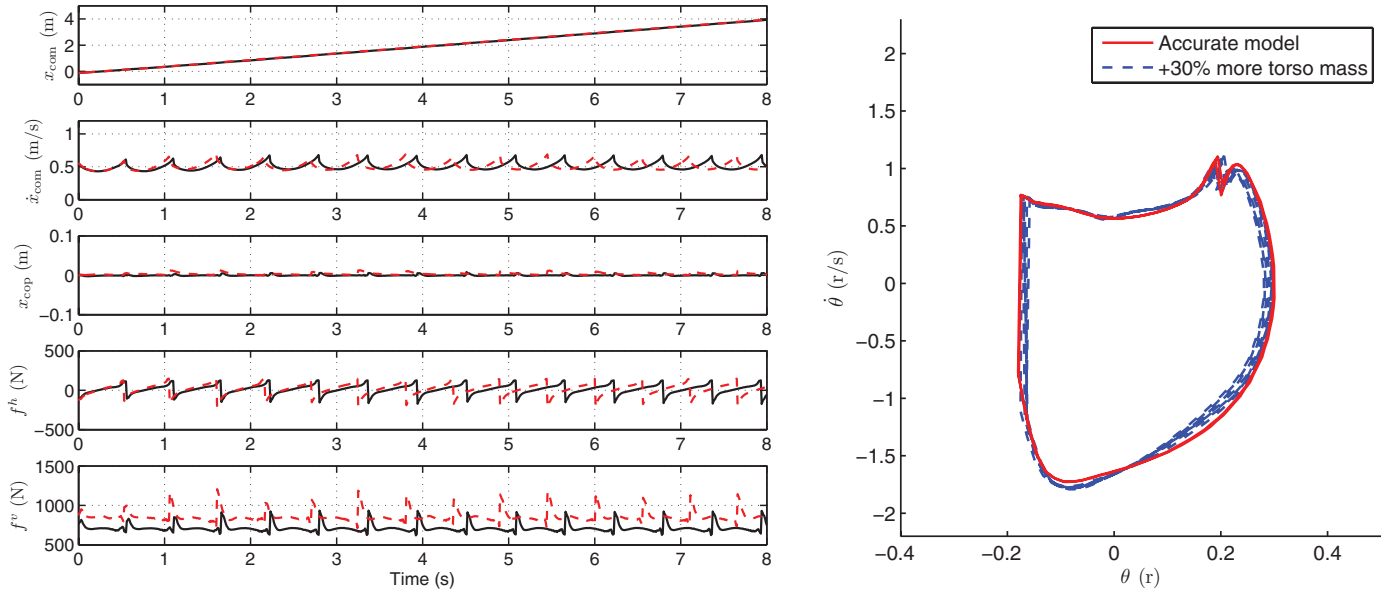
where f_x and f_y are, respectively, the horizontal and vertical components of the ground reaction force, k_x and k_y are spring



(a) (b)

Fig. 7. (Colour online) Walking on an incline of 5° as dashed lines. The solid lines are responses of walking control on a level ground. (a) The position and the velocity of the center of mass, the position of the CoP, and the ground reaction forces. (b) The phase portrait of the motion of one virtual leg.

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press



(a) (b)

Fig. 8. (Colour online) Walking control with a model error of 30% more torso mass as dashed lines. The solid lines are responses of walking control with an accurate model. (a) The position and the velocity of the center of mass, the position of CoP, and the ground reaction forces. (b) The phase portrait of the motion of one virtual leg.

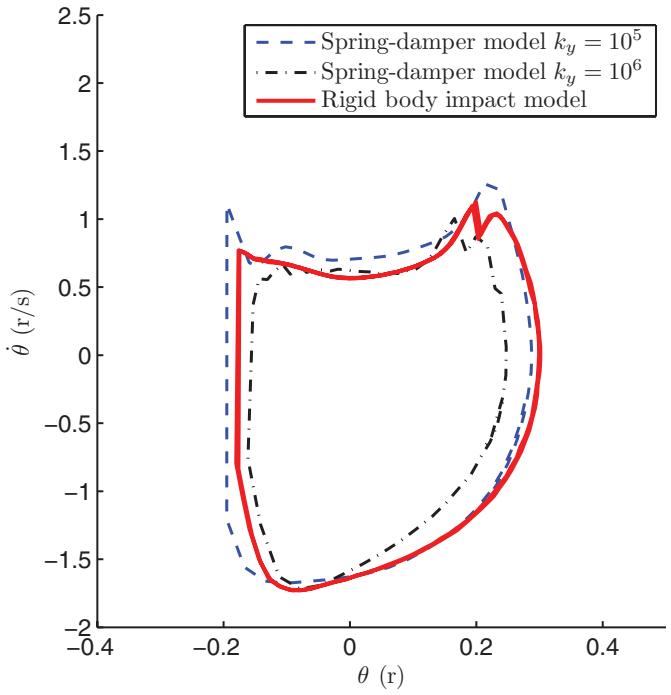


Fig. 9. (Colour online) Phase portraits of one virtual leg of normal walking using different foot–ground contact models.

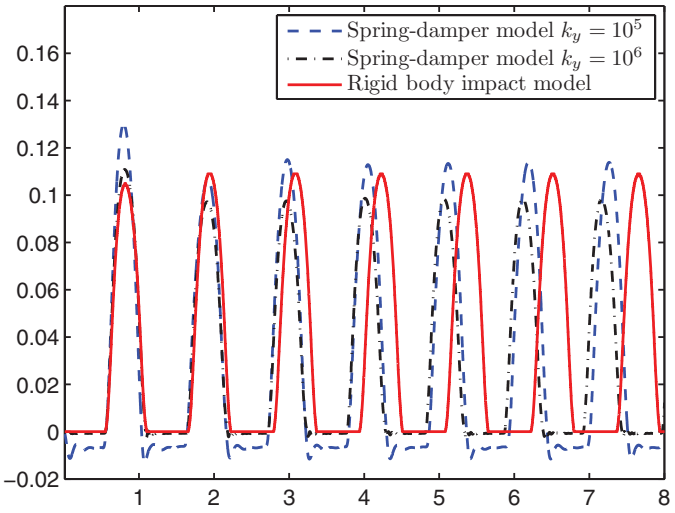


Fig. 10. (Colour online) Height of left foot during walking using different foot–ground contact models.

constants, c_x and c_y are damping coefficients, Δx and Δy are, respectively, the sliding distance and the depth under the ground of the leg's end, and $\dot{x}$ and $\dot{y}$ are the velocities of the leg's end. The proposed controller is evaluated using $k_x = 10^4$, $c_x = 10^3$, $c_y = 10^3$, and different k_y. As shown in Fig. 9, the proposed controller generated using the rigid body impact model transfers well to spring-damper models of different spring constants, although the robot trajectory converges to different limit cycles. For a specified foot, its height (as shown in Fig. 10) and the ground reaction force at it (as shown in Fig. 11) during walking are also different.

We compared the proposed controller with a trajectory tracking controller. For the trajectory tracking controller, each joint is driven by an independent PD + Feedforward controller, $u_j(i) = u_j^*(i) - k_{pj}(q_j(i) - q_j^*(i)) - k_{dj}(\dot{q}_j(i) - \dot{q}_j^*(i))$, where k_{pj} and k_{dj} are, respectively, the proportional and derivative gains of joint j. $u_j(i)$, $q_j(i)$, and $\dot{q}_i(i)$ are, respectively, the control torque, the angle, and the angular velocity of joint j at time index i, while $u_j^*(i)$, $q_j^*(i)$, and $\dot{q}_j^*(i)$ are the corresponding variables of the periodic steady-state trajectory. We optimized the total cost to find k_{pj} and k_{dj} and compared the minimized total cost over 6 s with that of the proposed controller. All the following simulations

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press

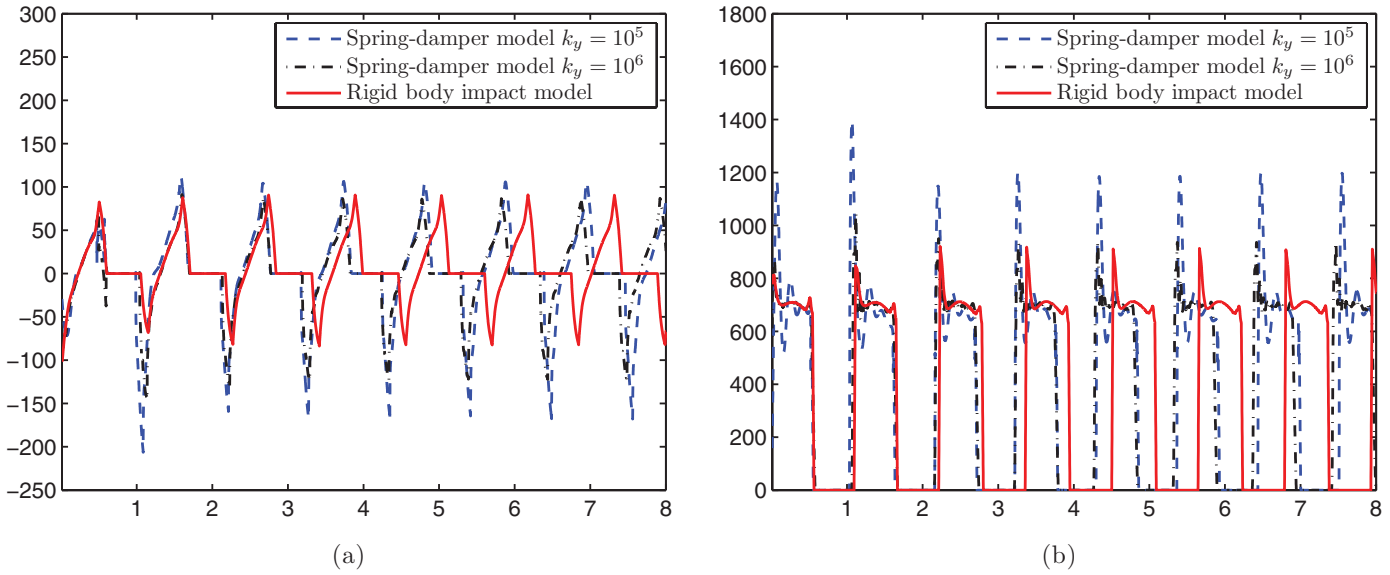


Fig. 11. (Colour online) The ground reaction force at the left foot during walking using different foot–ground contact models. (a) The horizontal component of the ground reaction force. (b) The vertical component of the ground reaction force.

start from the same initial state on the optimal trajectory and use the periodic steady-state trajectory only. For normal walking, the cost for the trajectory tracking controller is 3.661, while the corresponding cost for the proposed controller was 2.951. For walking in the presence of an impulsive perturbation of 5 Newton-seconds, the cost for the former was 21.679, the corresponding cost for the later was 3.697.

5. Conclusions

We use a parametric trajectory optimization method to solve the periodic steady-state trajectory in regular walking at a specified speed, which is then used as an initial trajectory for DDP to re-optimize and generate local models of the value function and the control law. We also use DDP to generate additional optimal trajectory segments to cover a larger portion of state space. By formulating the optimal control problem with an infinite time horizon, we get time-invariant locally linear models of the optimal control law, which are then used to construct a more global control law for biped walking. The results show lower cost from the proposed controller than a trajectory tracking controller using optimal gains.

Acknowledgments

This paper is based upon work supported by the National Natural Science Foundation of China Key Project (Grant No. 60935001) and in part by the US National Science Foundation (EEC-0540865, ECCS-0824077, and IIS-0964581) and the DARPA M3 program.

References

1. M. Vukobratovic, B. Borovac, D. Surla and D. Stokic, *Biped Locomotion: Dynamics, Stability, Control and Application* (*Scientific Fundamentals of Robotics*) (Springer, New York, 1990).
2. K. Hirai, M. Hirose, Y. Haikawa and T. Takenaka, "The Development of Honda Humanoid Robot," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation* vol. 2 (May 1998) pp. 1321–1326.
3. Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi and K. Tanie, "Planning walking patterns for a biped robot," *IEEE Trans. Robot. Autom.* **17**(3), 280–289 (2001).
4. S. Kajita and K. Tani, "Experimental study of biped dynamic walking," *IEEE Control Syst. Mag.* **16**(1), 13–19 (1996).
5. S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," *Proc. IEEE Int. Conf. Robot. Autom.* **2**, 1620–1626 (2003).
6. S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi and H. Hirukawa, "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* **1**, 239–246 (2001).
7. S. Kajita, O. Matsumoto and M. Saigo, "Real-time 3D walking pattern generation for a biped robot with telescopic legs," *Proc. IEEE Int. Conf. Robot. Autom.* **3**, 2299–2306 (2001).
8. L. Roussel, C. Canudas and A. Goswami, "Generation of Energy Optimal Complete Gait Cycles for Biped Robots," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium (May, 1998) pp. 2036–2041.
9. M. Hardt, "Multibody Dynamical Algorithms, Numerical Optimal Control, with Detailed Studies in the Control of Jet Engine Compressors and Biped Walking," *Ph.D. dissertation*, (San Diego, CA: University of California, 1999).
10. D. Djoudi, C. Chevallereau and Y. Aoustin, "Optimal Reference Motions for Walking of a Biped Robot," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain (Apr. 2005) pp. 2002–2007.
11. G. Bessonnet, P. Seguin and P. Sardain, "A parametric optimization approach to walking pattern synthesis," *Int. J. Robot. Res.* **24**(7), 523–536 (2005).
12. T. Saidouni and G. Bessonnet, "Generating globally optimised sagittal gait cycles of a biped robot," *Robotica* **21**(2), 199–210 (2003).
13. C. Chevallereau and Y. Aoustin, "Optimal reference trajectories for walking and running of a biped robot," *Robotica* **19**(5), 557–569 (2001).
14. C. Chevallereau, D. Djoudi and J. Grizzle, "Stable bipedal walking with foot rotation through direct regulation of the

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press

zero moment point," *IEEE Trans. Robot.* **24**(2), 390–401 (Apr. 2008).
15. K. Loffler, M. Gienger, F. Pfeiffer and H. Ulbrich, "Sensors and control concept of a biped robot," *IEEE Trans. Ind. Electron.* **51**(5), 972–980 (2004).
16. T. Komura, H. Leung, S. Kudoh and J. Kuffner, "A Feedback Controller for Biped Humanoids that Can Counteract Large Perturbations During Gait," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain (Apr. 2005) pp. 1989–1995.
17. P. B. Wieber and C. Chevallereau, "Online adaptation of reference trajectories for the control of walking systems," *Robot. Auton. Syst.* **54**(7), 559–566 (2006).
18. R. Bellman, *Dynamic Programming* (Princeton University Press, Princeton, NJ, 1957).
19. E. C. Whitman and C. G. Atkeson, "Control of Instantaneously Coupled Systems Applied to Humanoid Walking," *Proceedings of the IEEE International Conference on Humanoid Robotics*, Nashville, TN (Dec. 2010).
20. T. Mandersloot, M. Wisse and C. G. Atkeson, "Controlling Velocity in Bipedal Walking: A Dynamic Programming Approach," **In**: *Proceedings of the IEEE International Conference on Humanoid Robots*, Genova, Italy (Dec. 2006) pp. 124–130.
21. M. Stilman, C. G. Atkeson, J. Kuffner and G. Zeglin, "Dynamic Programming in Reduced Dimensional Spaces: Dynamic Planning for Robust Biped Locomotion," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain (Apr. 2005) pp. 2399–2404.
22. A. E. Bryson and Y. C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control* (Hemisphere, Washington, DC, 1975).
23. P. Dyer and S. R. McReynolds, *The Computation and Theory of Optimal Control* (Academic Press, Maryland Heights, MO, 1970).
24. D. H. Jacobson and D. Q. Mayne, *Modern Analytic and Computational Methods in Science and Mathematics*, Vol. 24: *Differential Dynamic Programming*. (Elsevier, Maryland Heights, MO, 1970).
25. J. Morimoto and C. Atkeson, "Minimax Differential Dynamic Programming: An Application to Robust Biped Walking," *Proceedings of the Sixteenth Annual Conference on Advances in the Neural Information Processing Systems*, Vancouver, Canada (2002).
26. C. G. Atkeson and J. Morimoto, "Nonparametric Representation of Policies and Value Functions: A Trajectory-Based Approach," **In**: *Proceedings of the Annual Conference on Advances in Neural Information Processing Systems* (2003) pp. 1643–1650.
27. Y. Tassa, T. Erez and W. Smart, "Receding horizon differential dynamic programming," *Adv. Neural Inf. Proc. Syst.* **20**, 1465–1472 (2008).
28. T. Erez, Y. Tassa and E. Todorov, "Infinite-Horizon Model Predictive Control for Periodic Tasks with Contacts," *Proceedings of Robotics: Science and Systems*, University of Southern California Los Angeles, CA, USA (2011).
29. C. Liu, C. G. Atkeson and J. Su, "Neighboring optimal control for periodic tasks for systems with discontinuous dynamics," *Sci. China Inf. Sci.* **54**(3), 653–663 (2011).
30. C. Liu and J. Su, "Biped Walking Control Using Offline and Online Optimization," *Proceedings of the* 30*th Chinese Control Conference*, Yantai, China (2011).
31. J. T. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming* (Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001).
32. C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *J. Guid. Control Dyn.* **10**(4), 338–342 (1987).
33. O. Von Stryk, "Numerical solution of optimal control problems by direct collocation," *Optim. control Calculus Variations Optim. Control Theory Numer. Methods* **111**(4), 129–143 (1993).
34. G. Elnagar, M. Kazemi and M. Razzaghi, "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE Trans. Autom. Control* **40**(10), 1793–1796 (Oct. 1995).
35. J. Vlassenbroeck and R. Van Dooren, "A Chebyshev technique for solving nonlinear optimal control problems," *IEEE Trans. Autom. Control* **33**(4), 333–340 (Apr. 1988).
36. P. E. Gill, W. Murray and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM J. Optim.* **12**, 979–1006 (1997).
37. O. V. Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Ann. Oper. Res.* **37**(1), 357–373 (1992).
38. C. G. Atkeson and B. Stephens, "Random sampling of states in dynamic programming," *IEEE Trans. Syst. Man Cybern.* B **38**(4), 924–929 (2008).
39. M. Stolle and C. G. Atkeson, "Finding and transferring policies using stored behaviors," *Auton. Robots* **29**(2), 169–200 (2010).
40. M. Breitner, "Robust optimal onboard reentry guidance of a space shuttle: dynamic game approach and guidance synthesis with neural networks," *J. Optim. Theory Appl.* **107**(3), 481–503 (2000).
41. J. J. Murray, C. Cox, G. G. Lendaris and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst. Man Cybern.* C **32**(2), 140–153 (2002).
42. A. Safonova and J. K. Hodgins, "Construction and optimal search of interpolated motion graphs," *ACM Trans. Graph.* **26**(3), 106 (2007).
43. K. Nishiwaki, T. Sugihara, S. Kagami, M. Inaba and H. Inoue, "Online Mixture and Connection of Basic Motions for Humanoid Walking Control by Footprint Specification," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea (May 2001) pp. 4110–4115.
44. J. Denk and G. Schmidt, "Synthesis of Walking Primitive Databases for Biped Robots in 3D-Environments," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan (Sep. 2003) pp. 1343–1349.
45. L. E. Kavraki, P. Svestka, J. C. Latombe and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.* **12**(4), 566–580 (1996).
46. C. G. Atkeson, "Using local trajectory optimizers to speed up global optimization in dynamic programming," **In**: *Advances in Neural Information Processing Systems*, vol. 6 (J. D. Cowan, G. Tesauro, and J. Alspector, eds.). (Morgan Kaufmann, Massachusetts, 1994) pp. 663–670.
47. R. Tedrake, "LQR-trees: Feedback Motion Planning on Sparse Randomized Trees," *Proceedings of Robotics: Science and Systems*, Seattle, USA (Jun. 2009).
48. P. Reist and R. Tedrake, "Simulation-Based LQR-Trees with Input and State Constraints," *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, USA (May 2010).
49. C. Liu and C. G. Atkeson, "Standing Balance Control Using a Trajectory Library," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, St. Louis, USA (Oct. 2009) pp. 3031–3036.
50. D. Bentivegna, C. Atkeson and J.-Y. Kim, "Compliant Control of a Hydraulic Humanoid Joint," **In**: *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Pittsburgh, PA, USA (Nov. 2007) pp. 483–489.
51. J. Grizzle, G. Abba and F. Plestan, "Asymptotically stable walking for biped robots: Analysis via systems with impulse effects," *IEEE Trans. Autom. Control* **46**(1), 51–64 (2001).

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press

52. L.-Z. Liao and C. Shoemaker, "Convergence in unconstrained discrete-time differential dynamic programming," *IEEE Trans. Autom. Control* **36**(6), 692 –706 (Jun. 1991).
53. Y. Tassa, "Theory and Implementation of Bio-Mimetic Motor Controllers," *Ph.D. dissertation*, Hebrew University of Jerusalem. Available at: URL: http://www.cs.washington.edu/homes/todorov/papers/TassaThesis.pdf (Feb. 2011).
54. M. Stolle and C. G. Atkeson, "Policies Based on Trajectory Libraries," *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, FL, USA (May 2006).

https://doi.org/10.1017/S0263574712000203 Published online by Cambridge University Press