SCIENCE CHINA
Information Sciences

• RESEARCH PAPERS •

March 2011 Vol. 54 No. 3: 653–663
doi: 10.1007/s11432-011-4185-z

Neighboring optimal control for periodic tasks for systems with discontinuous dynamics

LIU ChengGang[1]*, ATKESON Christopher G.[2] & SU JianBo[1]

[1]*Department of Automation, Shanghai Jiao Tong University, Shanghai* 200240*, China;*
[2]*Robotics Institute, Carnegie Mellon University, Pittsburgh* 15213*, PA, USA*

Received May 5, 2010; accepted November 30, 2010

Abstract We propose a trajectory-based optimal control method for periodic tasks for systems with discontinuous dynamics. A general method, dynamic programming, suffers from the problem of dimensionality. We use local models of the optimal control law to construct a local controller. We combine a parametric trajectory optimization method and differential dynamic programming (DDP) to find the optimal periodic trajectory in a periodic task. By formulating the optimal control problem with an infinite time horizon, DDP generates time-invariant local models of the optimal control law. For DDP, the value function at dynamics discontinuities is approximated by a second order Taylor series. The utility of the proposed method is evaluated using simulated walking control of a five-link biped robot. The results show lower torques and more robustness from the proposed controller than a PD servo controller.

Keywords trajectory optimization, differential dynamic programming, optimal control, biped walking control

Citation Liu C G, Atkeson Christopher G., Su J B. Neighboring optimal control for periodic tasks for systems with discontinuous dynamics. Sci China Inf Sci, 2011, 54: 653–663, doi: 10.1007/s11432-011-4185-z

1 Introduction

Many dynamic systems are subject to discontinuities in state at certain instants in time. Examples of such systems include a bouncing ball, rocket stage separation, and hopping, walking, or running mechanisms with intermittent ground contact. System dynamics of the type described above can be formulated as nonlinear systems with discontinuous dynamics. The purpose of this paper is to propose an optimal control method for periodic tasks for such systems using biped walking control as an example. The control of biped walking remains one of the most difficult research problems in robotics. It is a challenge due to high dimensionality, nonlinearity, the impact between the foot and the ground, and constraints on kinematics and dynamics, such as joint limitations, the foot clearance requirement, and the foot-ground contact conditions.

For a nonlinear system, a general method to find an optimal control law is dynamic programming [1]. For systems of continuous state and continuous action, a typical implementation of dynamic programming discretizes the state on a grid over the entire state space and stores the value function (the minimal cost to get to the goal) and/or the control law on the grid [2]. As a result, the computation and even the storage of the optimal control law becomes difficult when the dimension of state is high [1, 2]. Differential

*Corresponding author (email: cgliu2008@gmail.com)

© Science China Press and Springer-Verlag Berlin Heidelberg 2011

dynamic programming (DDP) is a local version of dynamic programming [3, 4]. It takes advantage of the Bellman equation, but applies the principle of optimality in the neighborhood of a given 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.

DDP is a gradient based trajectory optimization method and a good starting trajectory is important for its convergence to an optimum especially when the optimization problem is highly nonlinear and constrained. We take advantage of a parametric trajectory optimization method to find initial cyclic trajectories in periodic tasks such as walking. These trajectories are used as starting trajectories for DDP to re-optimize. A class of parametric trajectory optimization methods take the state and the control variables or the coefficients of their function approximations as optimization variables, the system dynamics as constraints, and minimize the cost function directly [5, 6]. The optimal control problem is then cast as a nonlinear optimization problem, which can be solved by general nonlinear programming methods, such as sequential quadratic programming (SQP) [7]. Parametric trajectory optimization methods are good at handling constraints and have the advantage of a wider range of convergence than other trajectory optimization methods.

In our approach, we use local models of the control law. Many efforts have been made to use trajectories and local models to represent feedback control laws [2, 8–15]. In [8] a trajectory library is used to establish a global control law for a marble maze task. In [9], Receding Horizon DDP was proposed to generate time-invariant local controllers. A trajectory library was used to synthesize a global controller for a simulated multi-link swimming robot. Ref. [11] created sets of locally optimized trajectories to handle changes to the system dynamics. In [15], locally-valid LQR controllers were used to construct a nonlinear feedback policy. Local models and local trajectory optimizers were used to accelerate global optimization in dynamic programming [16, 17].

We demonstrate the utility of our approach using simulated walking control of a five-link biped robot in the sagittal plane. By far, 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 ZMP (zero moment point) [18]. Emphasis is placed on enlarging the stability margin during gait planning [19, 20]. Trajectories can also be computed based on the LIPM (linear inverted pendulum model) [21]. Trajectory optimization of various cost criteria has been used to generate reference trajectories [22, 23]. Feedback control methods, such as PID controllers, computed torque, and ZMP feedback control are used to track the reference trajectories [19, 24, 25]. Control methods that do not rely on pre-computed trajectories have also been investigated [26–30].

We summarize the contributions of the present work: 1) We combine a parametric trajectory optimization method and DDP to find the optimal periodic trajectory in a periodic task. 2) We construct a controller from many local models of the optimal control law, which are byproducts of our trajectory optimization process. 3) For DDP to generate time-invariant local models, we formulate the optimal control problem with an infinite time horizon and approximate the value function at dynamics discontinuities by a second-order Taylor series. This paper is organized as follows: in section 2, the optimal control problem of periodic systems with discontinuous dynamics is formulated. Then, a parametric trajectory optimization method for periodic tasks is summarized, which solves for a periodic trajectory as a starting trajectory for DDP. Differential dynamic programming with an infinite time horizon is then outlined, which generates time-invariant linear local models of the optimal control law in the neighborhood the optimal trajectory. In section 3, the hybrid dynamics of a five-link biped robot walking in the sagittal plane is described. The optimization criteria and constraints are also introduced. In section 4, simulation results are presented, which demonstrate the validity and the utility of the proposed method. Conclusions and future work are discussed in section 5.

2 Control using trajectories and local models

2.1 Problem formulation

In the present paper we are interested in dynamic systems determined by ordinary differential equations with discontinuous dynamics. The class of systems with discontinuous dynamics under investigation can be described by equations of the form

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}), \qquad \boldsymbol{x} \notin \mathcal{S}, \tag{1}$$

$$\boldsymbol{x}^+ = H(\boldsymbol{x}^-), \qquad \boldsymbol{x}^- \in \mathcal{S}, \tag{2}$$

where $\boldsymbol{x}^- = \lim_{\tau \to t^-} \boldsymbol{x}(\tau)$ is the value of the state "just before the dynamics discontinuity", $\boldsymbol{x}^+ = \lim_{\tau \to t^+} \boldsymbol{x}(\tau)$ is the value of the state "just after the dynamics discontinuity", $H(\cdot)$ are the dynamics equations at the discontinuity, and $\mathcal{S}$ is a set of states. The system evolves according to ordinary differential equations until the state enters set $\mathcal{S}$. The dynamics equations at the discontinuity lead to a new initial state from which the dynamic system evolves until the next dynamics discontinuity.

In infinite horizon optimal control problems, the cost functional should include a discount factor to remain finite:

$$J := \int_0^\infty \mathrm{e}^{-\beta t} c(\boldsymbol{x}(t), \boldsymbol{u}(t)) \mathrm{d}t, \tag{3}$$

where $\beta > 0$ is the discount rate and $c(\cdot, \cdot)$ is the cost rate function. A state-feedback control law, $\boldsymbol{u} = \boldsymbol{u}^*(\boldsymbol{x})$, is optimal if the cost functional J is minimized by applying it. Constraints on the state and on the control,

$$b(\boldsymbol{x}, \boldsymbol{u}, t) \leqslant 0 \tag{4}$$

may be applied. In periodic tasks, there exist periodic trajectories that have translational symmetry:

$$\boldsymbol{x}(t + T) = \boldsymbol{x}(t), \tag{5}$$

where $T > 0$ is the period.

2.2 Periodic trajectory optimization

Considering the translational symmetry of periodic trajectories, the infinite horizon cost functional becomes

$$J = \int_0^\infty \mathrm{e}^{-\beta t} c(\boldsymbol{x}(t), \boldsymbol{u}(t)) \mathrm{d}t = (1 + \mathrm{e}^{-\beta T} + \mathrm{e}^{-\beta 2T} + \cdots) \int_0^T \mathrm{e}^{-\beta t} c(\boldsymbol{x}(t), \boldsymbol{u}(t)) \mathrm{d}t. \tag{6}$$

The power series converges to a limit,

$$1 + \mathrm{e}^{-\beta T} + \mathrm{e}^{-\beta 2T} + \cdots \to \frac{1}{1 - \mathrm{e}^{-\beta T}}. \tag{7}$$

Then

$$J = \frac{1}{1 - \mathrm{e}^{-\beta T}} \int_0^T \mathrm{e}^{-\beta t} c(\boldsymbol{x}(t), \boldsymbol{u}(t)) \mathrm{d}t. \tag{8}$$

We assume the dynamics discontinuity is traversed M times in one period. Let $0 = t_0 < t_1 \cdots < t_{M-1} < t_M = T$ denote the time instants when a dynamics discontinuity is crossed. Then

$$\begin{aligned} J = \frac{1}{1 - \mathrm{e}^{-\beta T}} \bigg[& \int_0^{t_1} \mathrm{e}^{-\beta t} c(\boldsymbol{x}(t), \boldsymbol{u}(t)) \mathrm{d}t + \mathrm{e}^{-\beta t_1} \phi(\boldsymbol{x}(t_1)) + \int_{t_1}^{t_2} \mathrm{e}^{-\beta t} c(\boldsymbol{x}(t), \boldsymbol{u}(t)) \mathrm{d}t + \mathrm{e}^{-\beta t_2} \phi(\boldsymbol{x}(t_2)) + \cdots \\ & + \int_{t_{M-1}}^{T} \mathrm{e}^{-\beta t} c(\boldsymbol{x}(t), \boldsymbol{u}(t)) \mathrm{d}t + \mathrm{e}^{-\beta T} \phi(\boldsymbol{x}(T)) \bigg], \end{aligned} \tag{9}$$

where $\phi(\cdot)$ is the cost at discontinuities.

We use a parametric trajectory optimization method to find an optimal periodic trajectory [5]. It uniformly divides each time interval $[t_{i-1}, t_i]$ into N_i time steps and uses a Runge-Kutta scheme to discretize the dynamics equations of (1). By taking the discrete-time dynamics equations at each time step as constraints and treating the state and the control variables at each time step and $t_1, \ldots, t_M$ as optimization variables, the trajectory optimization problem is cast as a nonlinear parametric programming problem. Constraints on the states and the controls (4), and

$$H(\boldsymbol{x}(N_i, i)) = \boldsymbol{x}(0, i+1), \quad i = 1, 2, \ldots, M-1, \tag{10}$$

$$H(\boldsymbol{x}(N_M, M)) = \boldsymbol{x}(0, 1), \tag{11}$$

$$\boldsymbol{x}(N_i, i) \in \mathcal{S}, \quad i = 1, 2, \ldots, M, \tag{12}$$

where $\boldsymbol{x}(j, i)$ denote the value of $\boldsymbol{x}$ at jth time step of ith time interval, are applied. This nonlinear programming problem can be handled by SNOPT, which uses an SQP algorithm [7].

2.3 Neighboring optimal control

To generate linear local models of the optimal control law, we use differential dynamic programming (DDP), which is a second order local trajectory optimization method, to further refine a starting trajectory. Discrete time DDP takes advantage of the Bellman equation [3, 4],

$$V(\boldsymbol{x}, k) = \min_{\boldsymbol{u}}[L(\boldsymbol{x}, \boldsymbol{u}) + \lambda V(F(\boldsymbol{x}, \boldsymbol{u}), k+1)], \tag{13}$$

where $L(\cdot, \cdot)$ is the one step cost function, $F(\cdot, \cdot)$ is the discrete-time dynamics equation, and λ is a discount factor. $V(\boldsymbol{x}, k)$ describes the minimal cost-to-go when the optimal feedback control, $\boldsymbol{u} = \boldsymbol{u}^*(\boldsymbol{x})$, is applied. In optimal control problems with an infinite time horizon, $V(\boldsymbol{x}, k)$ becomes time invariant and a function only of state, $V(\boldsymbol{x})$.

Given a starting trajectory $\boldsymbol{x}^j(k)$, 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 includes a discount factor (λ) and is given by

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

$$Q_x = \lambda V_x F_x + L_x, \tag{15}$$

$$Q_u = \lambda V_x F_u + L_u, \tag{16}$$

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

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

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

$$\boldsymbol{K} = Q_{uu}^{-1} Q_{ux}, \tag{20}$$

$$\Delta \boldsymbol{u} = Q_{uu}^{-1} Q_u, \tag{21}$$

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

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

where subscripts x and u indicate partial derivatives [17]. These partial derivatives are computed as a backward sequence starting from the end of the trajectory. Assuming the dynamics equations at the discontinuity, $H(\cdot)$, are differentiable, a second order Taylor series approximation of the value function on one side of the discontinuity, $V(k+1)$, and a linear approximation of $H(\cdot)$ leads to an approximation of the value function on the other side, $V(k)$:

$$\begin{aligned} V(k) &= \phi(\boldsymbol{x}_k) + V(H(\boldsymbol{x}_k)) \\ &\approx \phi(\boldsymbol{x}_k) + V(k+1) + V_x(k+1)(H(\boldsymbol{x}_k) - \boldsymbol{x}_{k+1}) \\ &\quad + \frac{1}{2}(H(\boldsymbol{x}_k) - \boldsymbol{x}_{k+1})^{\mathrm{T}} V_{xx}(k+1)(H(\boldsymbol{x}_k) - \boldsymbol{x}_{k+1}). \end{aligned} \tag{24}$$

Therefore,

$$V_x(k) = \phi_x(\boldsymbol{x}_k) + V_x(k+1) H_x(\boldsymbol{x}_k) + (H(\boldsymbol{x}_k) - \boldsymbol{x}_{k+1})^{\mathrm{T}} V_{xx}(k+1) H_x(\boldsymbol{x}_k), \tag{25}$$

$$V_{xx}(k) = \phi_{xx}(\boldsymbol{x}_k) + H_x(\boldsymbol{x}_k)^{\mathrm{T}} V_{xx}(k+1) H_x(\boldsymbol{x}_k). \tag{26}$$

For a periodic trajectory, the value function evaluated on the last point can be approximated with that evaluated on the first point. A new updated initial state is given by

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

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

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

where $\epsilon \in (0, 1)$. These processes are repeated until convergence to a local optimum is obtained.

To handle constraints (4), (10), (11) and (12), we augment the one step cost function $L(\cdot,\cdot)$ and the cost at discontinuities $\phi(\cdot)$ with quadratic penalties on the constraint violations. In practice, the inversion of Q_{uu} must be conditioned. In the case of close to singular $\boldsymbol{Q}_{uu}$, a Levenberg-Marquardt-like scheme can be used. Also, the initial state update (27) and the control sequence update (28) are performed with an adaptive line search of ϵ. 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.

Byproducts of DDP are linear local models of the optimal control law along the optimal trajectory,

$$\boldsymbol{u}_k = \bar{\boldsymbol{u}}_k - \bar{\boldsymbol{K}}_k(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k), \tag{29}$$

where $\bar{\boldsymbol{u}}_k$ and $\bar{\boldsymbol{x}}_k$ are respectively the control and the state on the optimal trajectory, and $\bar{\boldsymbol{K}}_k$ is the corresponding gain matrix. We formulate the optimal control problem with an infinite time horizon so that the value functions and control laws are time invariant and functions only of state. Because the local models (29) are local approximations to the optimal control law in the neighborhood of an optimal trajectory, they are also time-invariant and functions only of state. Thus, we can construct a controller from these spatially localized local models. The simplest choice is to select the nearest Euclidean neighbor as

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

where $\boldsymbol{x}$ is the current state, $\bar{\boldsymbol{u}}$, $\bar{\boldsymbol{x}}$, and $\bar{\boldsymbol{K}}$ are respectively the control variables, the state variables, and the gain matrix of the nearest local model. Similar approach has been applied to control of vertical hopping in a hopping robot [17] and control of a simulated multi-link swimming robot [9].

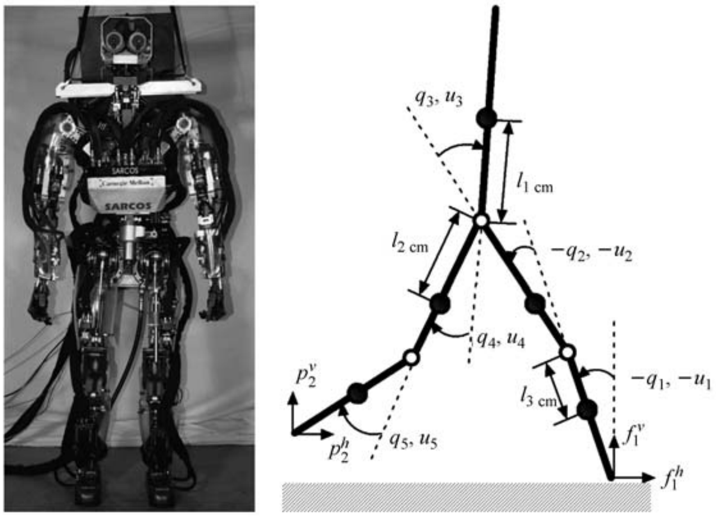
3 Experiments

The proposed method was evaluated using walking control of a five-link biped robot. The robot, as shown in Figure 1, is assumed to be planar and consists of a torso and two identical legs with knees. Furthermore, all body parts have mass, are rigid, and are connected with revolute joints. The robot is modeled on our Sarcos Primus System hydraulic humanoid robot and kinematic and dynamic parameters of the simulated robot are listed in Table 1. All walking cycles take place in the sagittal plane and consist of successive phases of single support and impact. During the single-support phase, the stance leg is touching the ground while the swing leg is not.

3.1 Single support model

In the single-support phase, a biped robot is modeled as a rotational joint open-chain manipulator with five links. The dynamic model of the robot during this phase has five degrees of freedom. Let $\boldsymbol{q} := (q_1, \ldots, q_5)^{\mathrm{T}}$ be the generalized coordinates describing the configuration of the robot depicted in Figure 1(b). Since only symmetric gaits are considered, 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

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}, \tag{31}$$



(a) (b)

Figure 1 (a) Sarcos Primus System hydraulic humanoid robot; (b) simplified structure of the biped robot used for our study.

Table 1 Physical parameters of the simulated robot

	Calf	Thigh	Torso
Mass (kg)	6.90	5.68	50.00
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

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{5\times5}$ is the inertia matrix, $\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^5$ is the vector of centrifugal, Coriolis, and gravity forces, $\boldsymbol{B}(\boldsymbol{q})$ is the matrix defining how the joint torques $\boldsymbol{u} := (u_1, \ldots, u_5)^{\mathrm{T}}$ enter the model. The second order system of (31) can be written in state space form as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{M}^{-1}(\boldsymbol{q})(-\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}) \end{bmatrix} = f(\boldsymbol{x}, \boldsymbol{u}), \quad 0 < t < T, \tag{32}$$

where $\boldsymbol{x} := (\boldsymbol{q}^{\mathrm{T}}, \dot{\boldsymbol{q}}^{\mathrm{T}})^{\mathrm{T}}$ and T is the duration of single-support phase.

3.2 Impact model

The impact between the swing leg and the ground is modeled as a contact between two rigid bodies. The assumptions are:

1. The revolute joints will be assumed to be ideal, that is, perfect elastic and no mechanical tolerance [31].
2. Centrifugal, Coriolis, and gravity forces ($\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}})$), and the joint torques are assumed to be smaller than the impulsive external forces and are neglected.
3. The impact is instantaneous. The impulsive forces due to the impact may result in an instantaneous change in the velocities, but there is no instantaneous change in the positions.
4. The contact of the swing leg end with the ground results in no rebound and no slipping of the swing leg.

The impact model results in a smooth map:

$$\boldsymbol{x}^+ = H(\boldsymbol{x}^-), \quad \boldsymbol{x}^- \in \mathcal{S}, \tag{33}$$

where $\boldsymbol{x}^-$ is the value of the state just prior to impact, $\boldsymbol{x}^+$ is the value of the state just after impact, $\mathcal{S} = \{\boldsymbol{x} | p_2^v(\boldsymbol{x}) = 0, \dot{p}_2^v(\boldsymbol{x}) < 0\}$ on flat ground (the definition of p_2^v is shown in Figure 1). The function

H represents the velocity change caused by impact and the relabeling of the robot's coordinates which makes the swing leg the stance leg (see Appendix).

The dynamics of overall biped robot is described by a nonlinear system with discontinuous dynamics. The robot mechanics system evolves according to the ordinary differential equations (32) until the state enters set $\mathcal{S}$ when impact occurs. The impact with the ground results in a rapid change in the velocity components of the state and the state components' relabeling. The result is a new initial state from which the robot model evolves until the next occurrence of impact.

3.3 Optimization criteria and constraints

The one step cost function $L(\boldsymbol{x}, \boldsymbol{u})$ we use penalizes the joint torques, $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{W}_u\boldsymbol{u}$, the walking speed error, $w_v(\dot{p}_3^h(\boldsymbol{x})-v_d)^2$, and the horizontal component of the ground reaction force, $w_r f_1^h(\boldsymbol{x})^2$, where $\boldsymbol{W}_u$, w_v, and w_r are penalty weights, v_d and $\dot{p}_3^h(\boldsymbol{x})$ are respectively the desired walking speed and the horizontal velocity of the hip. It 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 cost at discontinuities ϕ penalizes the horizontal velocity of the swing foot just before impact, $w_I \dot{p}_2^h(\boldsymbol{x}_N)^2$.

The robot's mechanism has some limitations on the range of joint angles, joint velocities, and joint torques. Moreover, the contact between the stance foot and the ground introduces some constraints. Although we do not model the feet, the horizontal position of the center of pressure (CoP) with respect to the ankle joint of the stance leg can be calculated by $x_{\mathrm{cop}} = -u_1/f_1^v$, where u_1 is the ankle torque of the stance leg. To keep the stance foot flat on the ground, the center of pressure (CoP) must be kept inside the foot-support region, $x_{\mathrm{cop}} \in \Omega$, where Ω denotes the foot-support region [18]. To avoid slipping, the ground reaction force must be kept inside the friction cone, $|f_1^h/f_1^v| \leqslant \mu$, where μ is the static friction coefficient. By penalizing the magnitudes of the ankle torque, u_1, and the horizontal component of the ground reaction force, f_1^h, we keep the center of pressure inside the foot-support region and the ground reaction force inside the friction cone.

4 Results

We use 20 grid points in time for SNOPT and 100 for DDP to optimize the periodic trajectory for walking speed $v_d = 0.5$ m/s. The parameters used by trajectory optimization are listed in Table 2. The stick diagram of the robot's motion with the optimal trajectory is shown in Figure 2(a), in which the configuration of the robot is drawn every 50 ms. The evolution of the value function on the optimal trajectory after each iteration is shown in Figure 2(b). The resultant feedback gains are shown in Figure 2(c). The resultant torques are shown in Figure 2(d).

The proposed controller was evaluated with a perturbation, which was a horizontal force (3000 Newtons) applied for 0.01 s (30 Newton-seconds impulse) at 3.0 s at the hip. The state of the robot was initialized on the optimal trajectory. As shown in Figure 3(b), our controller responds to the perturbation by driving the robot's trajectory back to the optimal trajectory (the solid line). The position and the velocity of the center of mass and the position of the CoP are shown in Figure 3(a). By penalizing the ankle torque of the stance leg, x_{cop} keeps inside the region of foot support, $[-0.1, 0.1]$.

The proposed controller was also evaluated using walking on an incline of 10. Simulation results show the proposed controller can still work. But as shown in Figure 4(b), the controller drives the robot's trajectory to a different limit cycle (the solid line). The center of pressure keeps inside the region of foot support, as shown in Figure 4(a).

We compared the proposed controller with a PD servo controller, in which each joint has a stiff PD controller to track the optimal trajectory. We use 1000 as the proportional gains and 10 as the derivative gains for all joints. We measured the sum of the squared torques, $\sum \boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}T$, over 6 s starting in a state on the optimal trajectory. For normal walking, the cost for the PD servo controller was 16270. The corresponding cost for the proposed controller was 4588. For walking in the presence of an impulsive perturbation of 5 Newton-seconds, the cost for the former was 22676, the corresponding cost for the later

Table 2 Parameters used by trajectory optimization

$\boldsymbol{W}_u$	$\mathrm{diag}(10^{-2}, 10^{-3}, 10^{-3}, 10^{-3}, 10^{-3})$	w_v	10^2	w_r	10^{-6}	h_d	0.05
w_I	10^{-1}	w_c	10^4	β	0.5	w_t	10

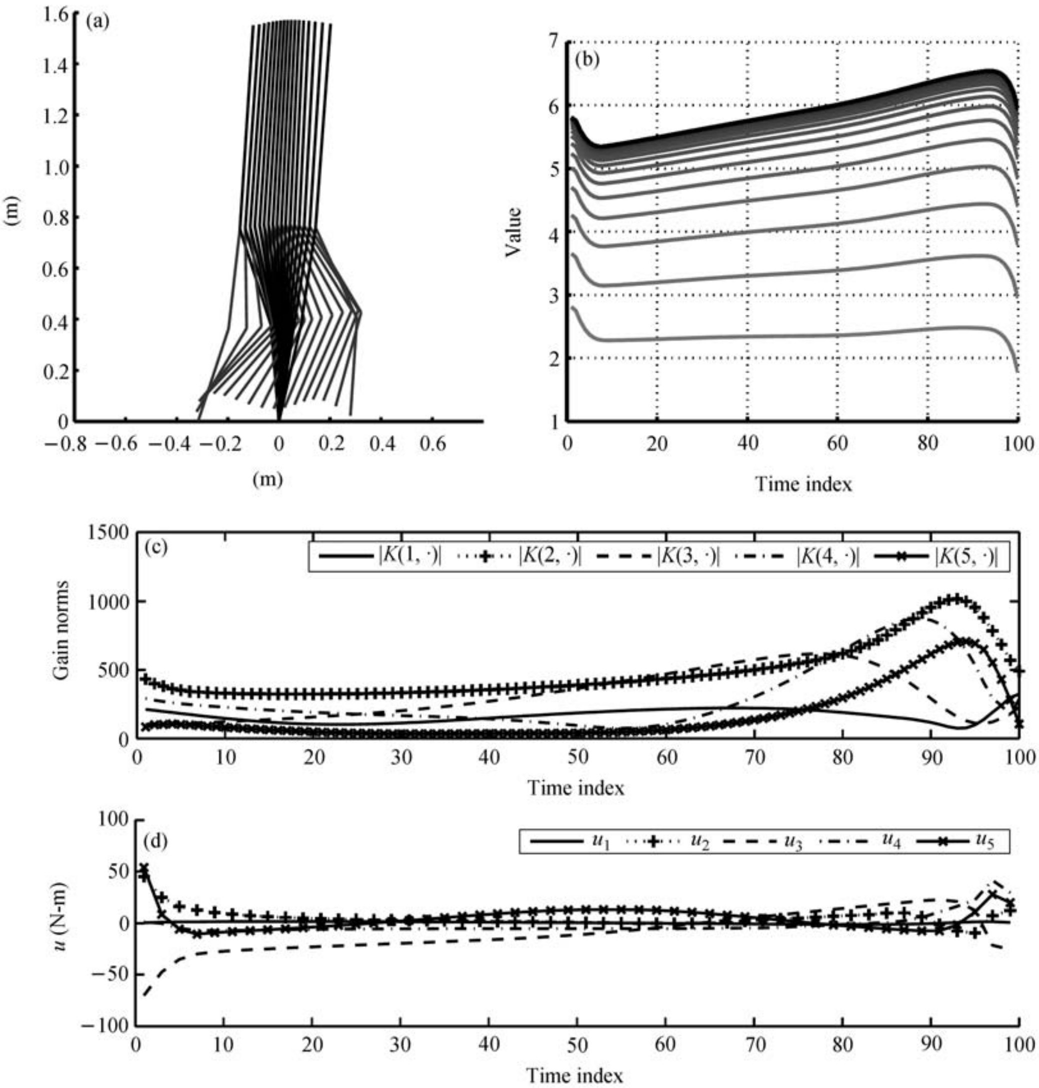


Figure 2 The optimal periodic trajectory at 0.5 m/s walking speed. (a) The stick diagram of the robot motion with the optimal trajectory. The configuration of the robot is drawn every 50 ms; (b) the evolution of the value function on the optimal trajectory after each iteration; (c) the norms of the feedback gains of each control on the grid points of the optimal trajectory; (d) the joint torques on grid points of the optimal trajectory. u_1, u_2, u_3 are those of the ankle, the knee, the hip of the stance leg, u_4 and u_5 are those of the hip and the knee of the swing leg.

was 4714. For walking on an incline of 3 degrees, the cost for the former was 29299, compared to 4901 for the latter. The PD servo controller falls down after an impulsive perturbation of larger than 5 Newton-seconds or on an incline of larger than 3 degrees. In contrast, the proposed controller is able to handle an impulsive perturbation of up to 36 Newton-seconds and an incline of up to 30 degrees.

5 Conclusions and future work

In the present paper we take advantage of a combination of a parametric trajectory optimization method and differential dynamics programming (DDP) to find an optimal cyclic trajectory in periodic tasks for

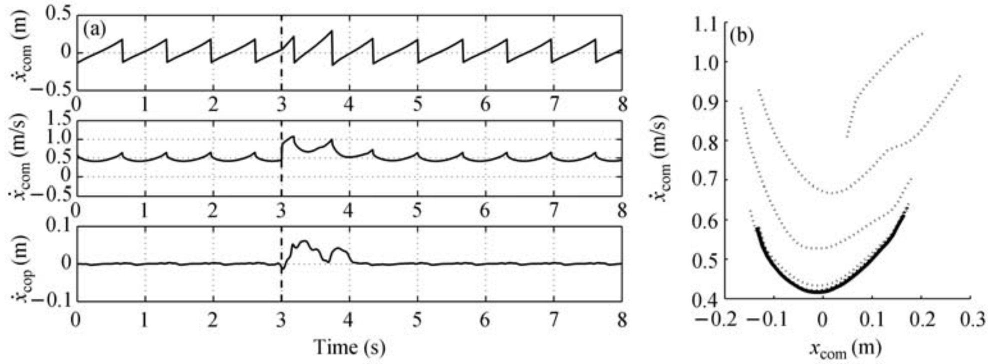


Figure 3 Response to a 30 Newton-seconds perturbation. (a) The horizontal position and velocity of the center of mass and the position of the CoP, which are all with respect to the ankle joint of the stance leg. The perturbation is applied at the dashed lines; (b) the phase portrait of the horizontal motion of the center of mass (dotted lines). The discontinuities are caused by the impulsive perturbation and the foot-ground contact. The solid line becomes a limit cycle.

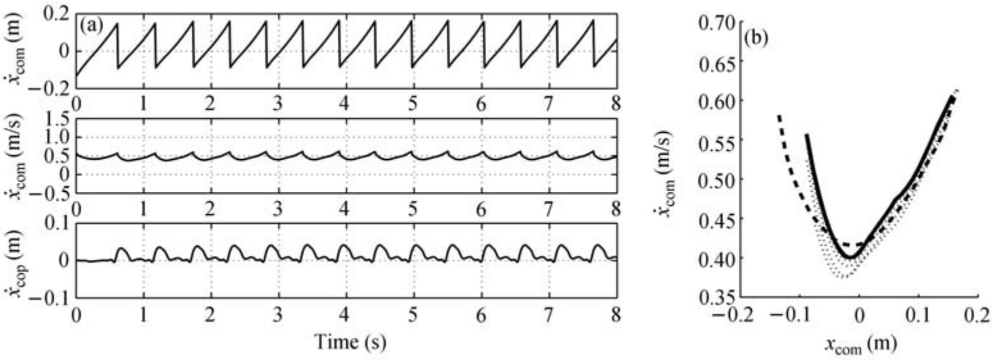


Figure 4 Walking on an incline of 10 degrees. (a) The horizontal position and the velocity of the center of mass and the position of the CoP, which are all with respect to the ankle joint of the stance leg; (b) the phase portrait of the horizontal motion of the center of mass (dotted lines). The discontinuities are caused by the foot-ground contact. The solid line becomes a new limit cycle. The dashed line is the limit cycle of normal walking.

systems with discontinuous dynamics. By formulating the optimal control problem with an infinite time horizon, we use DDP to get time-invariant local models of the optimal control law in the neighborhood of the optimal trajectory, which are used to construct a local controller. The utility of the proposed method is evaluated using biped walking control of a planar five-link robot, which is 10-dimensional and hard to solve using a tabular function approximation approach to dynamic programming.

The controller from a single trajectory may perform poorly far from the trajectory. To synthesize a more global controller, a library of optimal trajectories can be used. We have evaluated the library approach using standing balance control [32]. In our future work, we will use a library of optimal trajectories to construct a more global controller for periodic tasks. Actually implementing this algorithm on a robot is also expected.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 60935001), and in part by the National Science Foundation of US (Grant Nos. ECCS-0325383, EEC-0540865, ECCS-0824077, IIS-0964581).

References

1 Bellman R. Dynamic Programming. Princeton, NJ: Princeton Univ. Press, 1957

2 Larson R E. State Increment Dynamic Programming. New York: American Elsevier Pub. Co., 1968

3 Dyer P, McReynolds S R. The Computation and Theory of Optimal Control. San Diego, CA: Academic Press, 1970

4 Jacobson D H, Mayne D Q. Differential dynamic programming. In: Modern Analytic and Computational Methods in Science and Mathematics, No. 24. Elsevier, 1970

5 Betts J T. Practical Methods for Optimal Control Using Nonlinear Programming. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2001

6 Hargraves C R, Paris S W. Direct trajectory optimization using nonlinear programming and collocation. J Guid Control Dyn, 1987, 10: 338–342

7 Gill P E, Murray W, Saunders M A. SNOPT: An SQP algorithm for large-scale constrained optimization. SIAM J Opt, 1997, 12: 979–1006

8 Stolle M, Atkeson C G. Policies based on trajectory libraries. In: Proc. Int. Conf. Robot. Autom., Orlando, FL, USA, 2006. 3344–3349

9 Tassa Y, Erez T, Smart W. Receding horizon differential dynamic programming. In: Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2008, 20: 1465–1472

10 Grossman R L, Valsamis D, Qin X. Persistent stores and hybrid systems. In: Proc. Int. Conf. Decision and Control, New York, NY, USA, 1993. 2298–2302

11 Schierman J, Ward D, Hull J, et al. Integrated adaptive guidance and control for re-entry vehicles with flight test results. J Guid Control Dyn, 2004, 27: 975–988

12 Frazzoli E, Dahleh M, Feron E. Maneuver-based motion planning for nonlinear systems with symmetries. IEEE Trans Robot, 2005, 21: 1077–1091

13 Stolle M, Tappeiner H, Chestnutt J, et al. Transfer of policies based on trajectory libraries. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst, New York, NY, USA, 2007. 2981–2986

14 Safonova A, Hodgins J K. Construction and optimal search of interpolated motion graphs. ACM Trans Graph, 2007, 26: 106

15 Tedrake R, Manchester I R, Tobenkin M, et al. LQR-trees: Feedback motion planning via sums-of-squares Verification. Int J Rob Res, 2010, 29: 1038–1052

16 Atkeson C G. Using local trajectory optimizers to speed up global optimization in dynamic programming. In: Cowan J D, Tesauro G, Alspector J, eds. Advances in Neural Information Processing Systems. San Fransisco, CA: Morgan Kaufmann Publishers, Inc., 1994, 6: 663–670

17 Atkeson C G, Morimoto J. Nonparametric representation of policies and value functions: a trajectory-based approach. In: Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2003. 1643–1650

18 Vukobratović M, Borovac B, Surla D, et al. Biped locomotion: dynamics, stability, control and application. In: Scientific Fundamentals of Robotics. Berlin: Springer, 1990

19 Hirai K, Hirose M, Haikawa Y, et al. The development of Honda humanoid robot. In: Proc Int Conf Robot Autom, Leuven, Belgium, 1998, 2: 1321–1326

20 Huang Q, Yokoi K, Kajita S, et al. Planning walking patterns for a biped robot. IEEE Trans Robot Autom, 2001, 17: 280–289

21 Kajita S, Tani K. Experimental study of biped dynamic walking. IEEE Control Syst Mag, 1996, 16: 13–19

22 Djoudi D, Chevallereau C, Aoustin Y. Optimal reference motions for walking of a biped robot. In: Proc. Int. Conf. Robot. Autom., New York, NY, USA, 2005. 2002–2007

23 Bessonnet G, Seguin P, Sardain P. A parametric optimization approach to walking pattern synthesis. Int J Rob Res, 2005, 24: 523–536

24 Chevallereau C, Djoudi D, Grizzle J. Stable bipedal walking with foot rotation through direct regulation of the zero moment point. IEEE Trans Robot, 2008, 24: 390–401

25 Loffler K, Gienger M, Pfeiffer F, et al. Sensors and control concept of a biped robot. IEEE Trans Indust Electr, 2004, 51: 972–980

26 Westervelt E R, Grizzle J W, Koditschek D E. Hybrid zero dynamics of planar biped walkers. IEEE Trans Autom Contr, 2003, 48: 42–56

27 Raibert M H. Legged robots. Commun ACM, 1986, 29: 499–514

28 Pratt J, Pratt G. Intuitive control of a planar bipedal walking robot. In: Proc. IEEE Int. Conf. Robot. Autom., New York, NY, USA, 1998. 2014–2021

29 Dimitrov D, Ferreau H J, Wieber P B, et al. On the implementation of model predictive control for on-line walking pattern generation. In: Proc. IEEE Int. Conf. Robot. Autom., Pasadena, CA, USA, 2008. 2685–2690

30 Whitman E, Atkeson C G. Control of a walking biped using a combination of simple policies. In: Proc. Int. Conf. Humanoid Robots, Paris, 2009. 520-527

31 Moore M, Wilhelms J. Collision detection and response for computer animation. Comput Graph, 1988, 22: 289–298

32 Liu C G, Atkeson C G. Standing balance control using a trajectory library. In: Proc. Int. Conf. Intell. Robots Syst., St. Louis, 2009. 3031–3036

Appendix: Impact model

The impact between the swing leg and the ground is modeled as a contact between two rigid bodies. The stance foot may leave the ground during impact, so a 7-DoF robot model is used here. The generalized coordinates is extended by additional two coordinates as $\boldsymbol{q}_e = (q_1, \ldots, q_5, p_1^h, p_1^v)$, where p_1^h and p_1^v are the coordinates of the stance foot (as shown in Figure 1(b)). The 7-DoF dynamics is

$$\boldsymbol{M}_e(\boldsymbol{q}_e)\ddot{\boldsymbol{q}}_e + \boldsymbol{h}_e(\boldsymbol{q}_e, \dot{\boldsymbol{q}}_e) = \boldsymbol{B}_e\boldsymbol{u}_e + \boldsymbol{J}^{\mathrm{T}}\delta\boldsymbol{F}, \tag{A1}$$

where $\boldsymbol{u}_e := (u_1, \ldots, u_5, f_1^h, f_1^v)^{\mathrm{T}}$, $\delta\boldsymbol{F} := (f_2^h, f_2^v)^{\mathrm{T}}$, f_1^h, f_1^v, f_2^h, and f_2^v are the ground reaction forces during impact as shown in Figure 1(b). $\boldsymbol{J}$ is the Jacobian between the swing leg end and the extended coordinates,

$$\boldsymbol{J} := \frac{\partial \boldsymbol{p}_2}{\partial \boldsymbol{q}_e}, \tag{A2}$$

and $\boldsymbol{p}_2 := (p_2^h, p_2^v)^{\mathrm{T}}$. Assume the impact is instantaneous and the centrifugal, Coriolis, and gravity forces and the joint torques are assumed to be smaller than the impulsive external forces and are neglected. As a result, the impulsive forces due to the impact may result in an instantaneous change in the velocities, but there is no instantaneous change in the positions. Integrating eq. (A1) on the time interval of impact gives

$$\boldsymbol{M}_e(\boldsymbol{q}_e)(\dot{\boldsymbol{q}}_e^+ - \dot{\boldsymbol{q}}_e^-) = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{F}, \tag{A3}$$

where $\dot{\boldsymbol{q}}_e^+$ is the velocity just after impact, $\dot{\boldsymbol{q}}_e^-$ is the velocity just before impact, and $\boldsymbol{F} \in \mathbb{R}^2$ is the impulse at the contact point. Assuming the contact of the swing leg end with the ground results in no rebound and no slipping of the swing leg, we have

$$\dot{\boldsymbol{p}}_2 = \boldsymbol{J}\boldsymbol{q}_e^+ = 0. \tag{A4}$$

(A3) and (A4) give

$$\begin{bmatrix} \boldsymbol{M}_e(\boldsymbol{q}_e) & -\boldsymbol{J}^{\mathrm{T}} \\ \boldsymbol{J} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_e^+ \\ \boldsymbol{F} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_e(\boldsymbol{q}_e)\boldsymbol{q}_e^- \\ 0 \end{bmatrix}, \tag{A5}$$

which is a linear algebraic equation and its solution gives the joint velocities just after impact, $\dot{\boldsymbol{q}}^+$, and the impulse during impact, $\boldsymbol{F}$. The final impact model is

$$\boldsymbol{x}^+ = \begin{bmatrix} \boldsymbol{E} & 0 \\ 0 & \boldsymbol{E} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^- \\ \dot{\boldsymbol{q}}^+ \end{bmatrix} = H(\boldsymbol{x}^-), \tag{A6}$$

where $\boldsymbol{x}^-$ is the value of the state just prior to impact and $\boldsymbol{x}^+$ is the value of the state just after impact, $\boldsymbol{E}$ is a constant matrix such that $\boldsymbol{E}\boldsymbol{q}$ accounts for relabeling of the robot's coordinates which makes the swing leg the stance leg.