Alternative Formulations for Optimization-Based Human Gait Planning

Qian Wang [*], Yu-Jiang Xiang[†], Jasbir S. Arora[‡], and Karim Abdel- Malek[§]
Virtual Soldier Research Program/CCAD, College of Engineering, The University of Iowa, Iowa City, IA 52242

Simulating human motion is a complex problem due to redundancy of the human musculoskeletal system. The concept of task-based dynamic motion prediction using single- or multi-objective optimization techniques provides a viable approach for predicting dynamic gait motions of digital humans, subjected to basic physical and kinematical constraints. The task-based motion prediction is in fact a numerical optimal control problem. Alternative formulations for simulation of human gait motion are possible and can be solved by modern nonlinear optimization methods. Different ways to discretize the equations of motion are presented, namely finite difference, and Hermite and B-spline interpolations. The advantages and disadvantages of different formulations are discussed. Since the human gait simulation utilizes gradient-based optimization techniques, analytical gradients of objective and constraint functions are provided. A skeletal model for the lower body having 18 degrees of freedom is used to demonstrate the formulations, and is solved by a large-scale sparse nonlinear programming solver.

I. Introduction

Virtual human modelling and simulation has attracted considerable attention in recent years. Predicting a human gait motion is to solve a non-trivial dynamic problem, since joint angle rotations and torque profiles as well as ground reaction forces are all unknowns. In reality, humans can walk in an infinite variety of ways; therefore, there is no unique solution to the prediction of human gait motions. Several different research avenues have been explored in the literature. In the robotics field, a fast solution of the dynamics problem is needed to facilitate real-time motion and control. Thus, the basic information that needs to be generated is the motion trajectories of various segments and control torques. In the area of biomechanics, more natural and realistic human motions with complex musculoskeletal models have been studied and analyzed. Muscle tendon force, inner pressure, fatigue, and injury are all active areas of research.

The motion planning methods in robotics can be broadly categorized into those that are based on stability only without optimization of any performance measure, and those that use optimization to solve for optimal trajectories and torques. Motion capture, zero moment point (ZMP)-based trajectory generation, and inverted pendulum methods all belong to the first category. The motion capture approach is experiment-based; the motion of a subject is recorded by identifying the marker positions in the Cartesian coordinates. Then the joint angle trajectories are generated by using the recorded data and inverse kinematics. This approach is limited by the accuracy of the experimental data. Also, the simulated motion is subject-specific. Statistical methods are developed based on the pre-established motion database and they do not involve equations of motion[1,2]. The ZMP-based trajectory generation method enforces the stability of the mechanism in which leg motion of the biped and location of the ZMP are pre-planned[3,4]. Upper-body motion is generated to satisfy the ZMP trajectory; however, this may result in excessive upper-body motion. The key point of this approach is that the dynamics equations are used only to formulate the stability (ZMP) constraint condition rather than generation of the entire trajectory directly. Since additional equality constraints on gait parameters, such as hip/limb motion or the ZMP trajectory as a function of time are imposed, it is recommended to minimize the number of artificial constraints used since they may be too restrictive to adapt to changes in the mission goals, the anthropometric data, or the environment conditions. The inverted pendulum model is often used to solve the walking problem because biped walking can be treated as an

[*] Research Assistant; (qwang2@engineering.uiowa.edu).
[†] Research Assistant; (yujiang-xiang@uiowa.edu).
[‡] F. Wendell Miller Distinguished Professor of Engineering; Corresponding author (Jasbir-Arora@uiowa.edu).
[§] Professor and Director, CCAD, The University of Iowa (amalek@engineering.uiowa.edu).

inverted pendulum. The advantages of this method are its simplicity and fast solvable dynamics equations[5]. However, it also suffers from an inadequate dynamics model that cannot generate natural and realistic human motion.

Optimization-based trajectory generation is aimed at more realistic and natural humanoid motion. Many human-featured criteria can be simultaneously considered rather than only stability. For digital human simulations, the objective functions represent human performance measures, and optimization methods are used to solve for the feasible joint motion profiles that optimize the objective functions and satisfy the necessary constraints[6-9]. Lo *et al*. used quasi-Newton nonlinear programming techniques to determine the human motion that minimizes the actuating joint torques[10]. The design variables were the control points for the cubic B-spline approximation of joint angle profiles. Chevallereau and Aoustin planned a walking and running motion using the Pontryagin Maximum Principle to determine the coefficients of a polynomial approximation for profiles of the pelvis translations and joint angle rotations[6]. Saidouni and Bessonet solved for cyclic, symmetric gait motion of a nine degree of freedom (DOF) model that moves in the sagittal plane[8]. The control points for the B-spline curves along with the time durations for the gait stages are optimized to minimize the actuating torque energy. By adopting the time durations as design variables both the motion for the single support and for the double support are simultaneously optimized.

Skeletal models are used quite naturally in the robotics area. They have also been used in human locomotion modelling due to their relative simplicity and computational efficiency[6-9]. In biomechanics literature, optimization-based methods have been used to simulate human motion with complex musculoskeletal models[11-13]. Muscle groups are included in the model using Hill-type elements. The examples with musculoskeletal models include an 8 DOF model by Yamaguchi and Zajac[11] to restore unassisted natural gait to paraplegics and a model with 23 DOF and 54 muscles for normal symmetric walking on level ground by Anderson and Pandy[13]. Thelen *et al*. proposed a muscle control algorithm that used a static optimization method with feed-forward and feedback controls to obtain the desired kinematic trajectories of a musculoskeletal model[14]. The resulting simulation results matched well with the patterns of body-segmental displacements, ground-reaction forces, and muscle activations obtained from experiments. The forward dynamics optimization problem with such musculoskeletal models is typically posed to minimize the metabolic energy expenditure per unit distance travelled. A set of terminal posture constraints are often imposed to ensure repeatability of the gait cycle.

In the current paper, the Denavit-Hartenberg (DH) method[15] is used for modelling human skeletal links and dynamics. Meanwhile, various constraints are imposed to ensure the optimal motions satisfy maximum joint angle limit, no ground penetration, dynamic stability, and foot-point locations. The human body dynamics is based on a recursive formulation that can be used to calculate gradients efficiently. The equations of motion (EOMs) are discretized and finite dimensional approximation or parametric representation for the joint angles variables are defined, converting the simulation problem into a nonlinear programming (NLP) problem. Different discretization techniques are available. These techniques include finite difference, and piecewise polynomial and spline interpolations. In the current work, explicit integration of the equations of motion is avoided, which is very advantageous for large-scale problems. With these formulations, all the optimization constraints, i.e., limits on joint angles can be expressed explicitly in terms of the optimization variables. Therefore their gradient evaluations become simple. Some key features of the present work include: (1) three formulations based on different discretization methods are evaluated and compared for human gait planning simulations, (2) a recursive form of the equations of motion is discretized directly, and no EOMs are integrated. The present paper also uses a more recent SQP algorithm and associated software. One numerical example is optimized and its solutions are compared. Advantages and disadvantages of the formulations are discussed.

II. Human Motion Modelling

A skeletal model is used in this study. Human limbs and joints can be modelled as a series of linkages. The movements are generated by muscle forces, which act on the skeletal bones through lever arms thereby generating torques on joints. Before setting up the optimization formulation for solving the gait simulation, kinematics and dynamics analyses of the system need to be carried out. The DH method and recursive formulation are adopted for kinematics and dynamics analyses, respectively[16,17].

A. The DH method and Recursive Kinematics Formulation

The DH method is an approach for relating the position of a point in one coordinate system to another, by using transformation matrices[15]. In order to obtain a systematic representation of a serial kinematics chain, $\mathbf{q} \in R^d$ is defined as the vector of *d*-generalized coordinates, the joint angles. The position vector of a point of interest in the Cartesian space can be written in terms of the joint variables as $\mathbf{X} = \mathbf{X}(\mathbf{q})$, where $\mathbf{X}(\mathbf{q})$ can be obtained from the multiplication of the 4×4 homogeneous transformation matrices ${}^{i-1}\mathbf{T}_i$ relating coordinate frames i and i-1, represented by four parameters θ_i, d_i, α_i, and a_i, as shown in Figure 1.

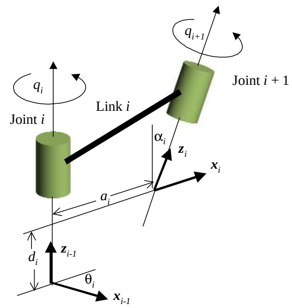


Figure 1. Joint coordinate systems

Let us define augmented 4×1 vectors ${}^0\mathbf{r}_d$ and $\mathbf{r}_d$ using the global Cartesian vector $\mathbf{X}(\mathbf{q})$ and the local Cartesian vector $\mathbf{X}_d$ as:

$$ {}^0\mathbf{r}_d = \begin{bmatrix} \mathbf{X}(\mathbf{q}) \\ 1 \end{bmatrix}, \ \mathbf{r}_d = \begin{bmatrix} \mathbf{X}_d \\ 1 \end{bmatrix} \tag{1} $$

where $\mathbf{X}_d$ is the position of the point with respect to the dth coordinate system. Using these vectors, ${}^0\mathbf{r}_d$ can be related to $\mathbf{r}_d$ (i.e., the global Cartesian vector $\mathbf{X}(\mathbf{q})$ can be expressed in terms of the local Cartesian vector $\mathbf{X}_d$) as:

$$ {}^0\mathbf{r}_d = {}^0\mathbf{T}_d(\mathbf{q})\mathbf{r}_d \tag{2} $$

where

$$ {}^0\mathbf{T}_d(\mathbf{q}) = \prod_{i=1}^{d} {}^{i-1}\mathbf{T}_i = {}^0\mathbf{T}_1(q_1)\,{}^1\mathbf{T}_2(q_2)\cdots{}^{d-1}\mathbf{T}_d(q_d) \tag{3} $$

According to the above analysis, we can define matrices $\mathbf{A}_j$, $\mathbf{B}_j$, and $\mathbf{C}_j$ as recursive position, velocity, and acceleration transformation matrices for the jth joint, respectively. Given the link transformation matrix ($\mathbf{T}_j$) and the kinematics state of each joint variable (q_j, $\dot{q}_j$ and $\ddot{q}_j$), then for j =1, d (i.e., an d degree of freedom chain), we have:

$$ \mathbf{A}_j = \mathbf{T}_1\mathbf{T}_2\mathbf{T}_3\cdots\mathbf{T}_j = \mathbf{A}_{j-1}\mathbf{T}_j \tag{4} $$

$$ \mathbf{B}_j = \dot{\mathbf{A}}_j = \mathbf{B}_{j-1}\mathbf{T}_j + \mathbf{A}_{j-1}\frac{\partial \mathbf{T}_j}{\partial q_j}\dot{q}_j \tag{5} $$

$$ \mathbf{C}_j = \dot{\mathbf{B}}_j = \ddot{\mathbf{A}}_j = \mathbf{C}_{j-1}\mathbf{T}_j + 2\mathbf{B}_{j-1}\frac{\partial \mathbf{T}_j}{\partial q_j}\dot{q}_j + \mathbf{A}_{j-1}\frac{\partial^2 \mathbf{T}_j}{\partial q_j^2}\dot{q}_j^2 + \mathbf{A}_{j-1}\frac{\partial \mathbf{T}_j}{\partial q_j}\ddot{q}_j \tag{6} $$

where $\mathbf{A}_0 = [\mathbf{I}]$ and $\mathbf{B}_0 = \mathbf{C}_0 = [\mathbf{0}]$. After obtaining all the transformation matrices $\mathbf{A}_j$, $\mathbf{B}_j$, and $\mathbf{C}_j$, the global position, velocity, and acceleration of a point in the Cartesian coordinates can be calculated as[18]

$$ {}^0\mathbf{r}_d = \mathbf{A}_d\mathbf{r}_d \tag{7} $$

$$ {}^0\dot{\mathbf{r}}_d = \mathbf{B}_d\mathbf{r}_d \tag{8} $$

$$ {}^0\ddot{\mathbf{r}}_d = \mathbf{C}_d\mathbf{r}_d \tag{9} $$

where $\mathbf{r}_d$ is the local coordinates of the point in the dth coordinate system.

B. Dynamics (Recursive Lagrangian Equations)

The general form of dynamic equations of motion is derived from energy principle. Based on the recursive kinematics analysis of the previous section, the backward recursion for the dynamic analysis is accomplished by defining transformation matrices $\mathbf{D}$ and $\mathbf{E}$ as follows[18]. Given the mass and inertia properties of each link, then the joint actuation forces/torques, τ_i, are computed for $i = d$ to 1 using

$$\tau_i = tr\left[\frac{\partial \mathbf{A}_i}{\partial q_i}\mathbf{D}_i\right] - \mathbf{g}^T \frac{\partial \mathbf{A}_i}{\partial q_i}\mathbf{E}_i \quad (10)$$

$$\mathbf{D}_i = \mathbf{J}_i \mathbf{C}_i^T + \mathbf{T}_{i+1}\mathbf{D}_{i+1} \quad (11)$$

$$\mathbf{E}_i = m_i\, {}^i\mathbf{r}_i + \mathbf{T}_{i+1}\mathbf{E}_{i+1} \quad (12)$$

where $\mathbf{D}_{d+1} = \mathbf{E}_{d+1} = [\mathbf{0}]$; $\mathbf{J}_i$ = inertia matrix for link *i;* m_i = mass of link *i;* $\mathbf{g}$ = gravity vector; ${}^i\mathbf{r}_i$ = location of the center of mass of link *i* in link *i* frame. $tr[\cdots]$ denotes the trace operation, and $(\cdots)^T$ denotes the matrix transpose. The first term in the torque expression denotes inertia and Coriolis torques; the second term denotes the torque due to gravity. Gradients information for all transformation matrix and torques with respect to state variables can be also evaluated in a recursive way as shown in Eqs. (4) to (12).

III. Human Gait Planning as an Optimization Problem

It has been shown that task-based human motion prediction is in fact a numerical optimal control problem (OCP)[19]. The basic optimal control problem is to determine unknown quantities such as joint angles and torques, to achieve certain goals (e.g., minimization of a performance measure function) while satisfying all the performance requirements or constraints.

A. Objective Function

A general objective function (performance index PI) for optimal control problem is defined as:

$$J = c_0(\mathbf{q}(T), T) + \int_0^T h_0(\boldsymbol{\tau}, \mathbf{q}, t)\, dt \quad (13)$$

where T is the total time interval considered. Note that the definition of the OCP problem contains a wide variety of control problems, such as minimum time, minimum control effort, trajectory tracking, and response constraints. The joint angles $\mathbf{q}$ and torques $\boldsymbol{\tau}$ are called *state variables* and *control variables* in the OCP problem. For digit human gait simulation, the objective functions represent human performance measures. Various performance measures of digital human have been developed in the literature[8,10,12,18,20].

B. Optimization Constraints

The optimization constraints consist of the equations of motion in Eqs. (10)-(12), and other time-dependent requirements, as

$$\mathbf{g}(\boldsymbol{\tau}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) \le \mathbf{0} \quad (14)$$

This type of constraints is the so-called dynamic or *point-wise* constraint, which needs to be satisfied at each point of the entire time interval $t \in [0, T]$. The other type of constraints are not functions of t; therefore, they can be treated easily in the optimization process. These constraints include the initial and final motion constraints. Five treatments of the point-wise constraints in Eq. (14) have been discussed in the literature[21]. Note that time-dependent constraints on the state variables can be imposed at the discrete time grid points. In dynamic gait simulations, the time-dependent inequality constraints in Eq. (14) may include the following basic constraints: (1) foot ground penetration, (2) foot strike position, (3) ZMP stability condition, (4) joint angle and torque limits, and others.

1. Foot ground penetration

Foot ground penetration in each phase (controls height of foot points as shown in Figure 2). For foot points without contact (triangular points in Figure 2), it is required that $x \ge 0$; for foot points with contact condition (circular points in Figure 2), the condition is $x = 0$, where x is the vertical coordinate of any point on a foot. It is seen that the number of foot ground penetration constraints depends on the total number of points on the two feet.

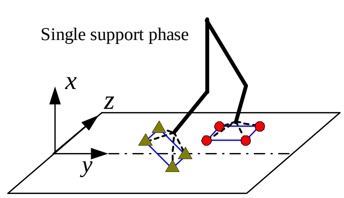


Figure 2. Foot ground penetration constraints

2. Foot strike position

Foot strike position can be used to determine walking direction and step length. As shown in Figure 3, the strike coordinates can be imposed as equality constraints.

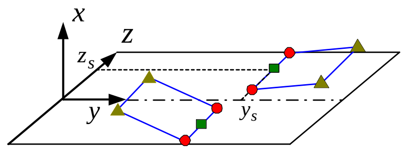


Figure 3. Foot strike position constraints

3. ZMP stability

The dynamic stability that refers to dynamic equilibrium of all the forces on a walking biped, is imposed by restricting the ZMP location within the boundary of foot support region (FSR). The ZMP concept dates back thirty-five years and has been re-examined by recent literature either for clarification or extension[22,23]. The ZMP coordinates and stability requirements are as follows:

$$y_{ZMP} = \frac{\sum_{i=1}^{ndof} m_i(\ddot{x}_i + g)y_i - m_i\ddot{y}_i x_i + I\ddot{\theta}_{iz}}{\sum_{i=1}^{ndof} m_i(\ddot{x}_i + g)} \in FSR \tag{15}$$

$$z_{ZMP} = \frac{\sum_{i=1}^{ndof} m_i(\ddot{x}_i + g)z_i - m_i\ddot{z}_i x_i + I\ddot{\theta}_{iy}}{\sum_{i=1}^{ndof} m_i(\ddot{x}_i + g)} \in FSR \tag{16}$$

where x_i, y_i, and z_i are the coordinates of the center of mass for *i*th link; m_i is the mass, I_i is the global inertia, and $\ddot{\theta}_i$ is the global angular acceleration of the *i*th link, respectively.

4. Joint angle and torque limits

The constraints on the joint angle profiles and torques are

$$\mathbf{q}^L \le \mathbf{q} \le \mathbf{q}^U, \ \boldsymbol{\tau}^L \le \boldsymbol{\tau} \le \boldsymbol{\tau}^U \tag{17}$$

where $\mathbf{q}^L, \mathbf{q}^U$ and $\boldsymbol{\tau}^L, \boldsymbol{\tau}^U$ are the lower and upper bounds for the joint displacements and torques, respectively. Note that the joint torque limits $\boldsymbol{\tau}^L$ and $\boldsymbol{\tau}^U$ are sometimes difficult to obtain; therefore, a viable way is to treat them as a part of the objective function. This constitutes a so-called minimum control effort problem, which has no particular difficulty for solution.

IV. Solution Techniques of Optimal Control Problem

The optimal control problems find the best solution that achieves user-specified objective through forward or inverse dynamics combined with optimization. Several different ways based on numerical optimization to solve the OCP problem are available, namely state variable elimination (forward dynamics), direct collocation, and differential inclusion (inverse dynamics).

A. State Variable Elimination

Forward dynamics starts with initial conditions and known forces and solves for unknown joint displacements by numerical integration[13,24,25]. The process of integrating forces over time intervals to obtain walking motions can itself be computationally intensive. Design sensitivity analysis is also needed in this formulation, since the joint angle, velocities and accelerations are implicit functions of the optimization variables, the joint torques[12,20]. One way to deal with the computational intensity of forward dynamics formulations has been to use massively parallel algorithms and processing scheme[12].

B. Direct Collocation

This is the simultaneous formulation where both the joint torques and the joint profiles are treated as optimization variables. Therefore the equations of motion in Eqs. (10) to (12) are treated as equality constraints in the formulation. The direct collocation approach has been used in other engineering fields[26-31], and robotic or human motion planning[32]. Although there are a large number of variables in this formulation, the equations of motion are not required to be satisfied at each iteration of the optimization process. They only need to be satisfied at the final optimum point of the problem. There are two main advantages of this formulation for dynamic systems: (i) the equations of motion for the system need not be integrated explicitly, (ii) design sensitivity analysis of the systems is not needed since all the problem functions are explicit in terms of the variables. With these formulations, the optimization problem becomes large; i.e., the numbers of variables and constraints are large. However, the problem functions are quite sparse; i.e., each function depends on only a few variables. These sparse properties of the functions can be exploited in the optimization process.

C. Differential Inclusion

The inverse dynamic method, on the other hand, calculates unknown forces from joint displacement histories. The joint displacement histories associated with locomotion are determined using optimization methods[6,7]. Two important issues in such inverse dynamics frameworks are the human performance criteria and methods for approximating the joint trajectories. The work of Lo *et al*.[10], although it deals with human motions and tasks other than locomotion, provides a thorough description of an inverse dynamics framework for predicting human motions. Note that in this formulation, the equations of motion in Eqs. (10) to (12) are not integrated nor treated as equality constraints; they are automatically satisfied in the optimization process.

V. Discretization Techniques of Equations of Motion

Different discretization techniques for the equations of motion in Eqs. (10) to (12) are available. The general idea is to transfer the EOMs to an algebraic system of equations, the so-called defect equations, which need to be set to zero to enforce the EOMs. In the next section, some of these formulations are presented, and they are based on finite difference method, and Hermite and B-spline interpolations.

A. Central Difference (CD)

This is the perhaps the easiest way to discretize the system of dynamic equations. Some common methods include forward, backward difference, and central difference. In the central difference method, the joint velocity and acceleration vectors $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are written explicitly with respect to the joint angle vector $\mathbf{q}$, as follows:

$$\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_i(t_i) = \frac{\mathbf{q}_{i+1} - \mathbf{q}_{i-1}}{2\Delta t}, \qquad i = 0, N \tag{18}$$

$$\ddot{\mathbf{q}}_i = \ddot{\mathbf{q}}_i(t_i) = \frac{\mathbf{q}_{i+1} - 2\mathbf{q}_i + \mathbf{q}_{i-1}}{\Delta t^2}, \qquad i = 0, N \tag{19}$$

where Δt is the time interval ($\Delta t = T/N$). If some of the state variables, such as the joint angle displacements $\mathbf{q}$, are treated as variables in the optimization formulation, the implicit problem becomes explicit. Therefore special design sensitivity analysis procedures are not needed, when the direct collocation or inverse dynamic methods are employed. The finite difference approximations are used to replace the joint velocities and accelerations in terms of the angles. Note that Eqs. (18) and (19) are the additional equality constraints among the variables.

B. Piecewise Hermite Interpolation (Hermite)

The basic discretization scheme is as follows: the state variables $\mathbf{q}$ are chosen as continuous differentiable functions and piecewise defined as cubic polynomials between $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$, with the EOMs (10) to (12) satisfied at t_i and t_{i+1}. For $t \in [t_i, t_{i+1}]$, the approximation of state variables $\mathbf{q}$ is

$$\mathbf{q}_a(t) = \sum_{k=0}^{3} \mathbf{c}_k^i \left(\frac{t - t_i}{\Delta t} \right)^k \tag{20}$$

where

$$\begin{aligned} &\mathbf{c}_0^i = \mathbf{q}_i; \mathbf{c}_1^i = \Delta t \dot{\mathbf{q}}_i \\ &\mathbf{c}_2^i = -3\mathbf{q}_i - 2\Delta t \dot{\mathbf{q}}_i + 3\mathbf{q}_{i+1} - \Delta t \dot{\mathbf{q}}_{i+1}; \mathbf{c}_3^i = 2\mathbf{q}_i + \Delta t \dot{\mathbf{q}}_i - 2\mathbf{q}_{i+1} + \Delta t \dot{\mathbf{q}}_{i+1} \end{aligned} \tag{21}$$

where $\Delta t = t_{i+1} - t_i$. The approximation function (20) of the state variables must satisfy the EOMs (10)-(12) at the grid point t_i ($i = 0, N$). The time dependent constraints are satisfied at the grid points. The optimization problem is to determine $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\boldsymbol{\tau}$ to minimize the cost function of Eq. (13), subject to the discretized inequality constraints in Eq. (14) at the time grid point. Note that at the grid point $t = t_i$,

$$\ddot{\mathbf{q}}_i = \ddot{\mathbf{q}}_a(t_i) = \frac{2}{\Delta t^2}\left(-3\mathbf{q}_i + 3\mathbf{q}_{i+1} - 2\Delta t\dot{\mathbf{q}}_i - \Delta t\dot{\mathbf{q}}_{i+1}\right) \tag{22}$$

C. Cubic B-spline Interpolation (B-spline)

Since human motion trajectories are usually very smooth, cubic B-spline interpolation can be used. For the optimization problem, the entire time domain is discretized by B-spline curves, which are defined by a set of control points **P** and time grid points (knots) **t**. B-spline is a numerical interpolation method that has many important properties, such as continuity, differentiability, and local control[33]. These properties, especially differentiability and local control, make B-splines competent to represent joint angle trajectories, which require smoothness and flexibility. There are a number of ways to define the B-spline basis functions, and it is preferred to have an explicit polynomial form rather than in a recursive form. Let $T = \{t_0, t_1,.., t_m\}$ be a non-decreasing sequence of real numbers, i.e., $t_i \le t_{i+1},\ i = 0, .., m\text{-}1$. The t_i are called *knots*, and they are non-decreasingly spaced. A cubic B-spline is defined as

$$q(t) = \sum_{j=0}^{nct} N_{j,4}(t)P_j; \qquad 0 \le t \le T \tag{23}$$

where the $\{P_j\},\ j = 0, .., nct$ are the $(nct+1)$ control points, and the $\{N_{j,4}(t)\}$ are the cubic B-spline basis functions defined on the knot vector (($m+1$) knots). In reality each segment of the curve is defined by four control points. The *k*th derivatives of a cubic B-spine curve can be easily obtained from Eq. (23), since only the basis functions are functions of time. In this formulation, the control vector $\mathbf{P}$ for each DOF are chosen as the optimization variables. This formulation is to minimize the objective function in Eq. (13), subject to the inequality constraints in Eq. (14), as

$$\mathbf{g}(\boldsymbol{\tau}, \mathbf{P}, t) \le \mathbf{0} \tag{24}$$

D. Discussion of Formulations

Advantages and disadvantages of different formulations are listed in Table 1. Since the objective and constraint functions are all explicit in terms of the optimization variables in the direct collocation or differential inclusion formulations, the gradients of functions can be obtained easier than the state variable elimination method. Starting form a system of differential equations, approximations in the time domain for the state variables are set up and collocation can be enforced on certain time points in the direct collocation formulations. The equations of motion in Eqs. (10)-(12) do not need to be satisfied exactly at each iteration of the optimization process. They need to be satisfied only at the final solution point. This has advantage if instabilities occur or no solution exists for differential equations (DEs) for certain points in the design space. Also, unnecessary simulations of the system are avoided at intermediate designs, where it might be difficult to obtain a solution. The differential inclusion formulation does not need the integration of the dynamic equations, either. In these formulations, the system of DEs is directly discretized and imbedded into the optimization formulation. However, the error in the solution of DEs in state variable elimination formulations can be easily controlled, which is not an easy task for the other two formulations. Differential inclusion formulation includes fewer variables and needs less storage than direct collocation; therefore it has special advantage. Note that differential inclusion formulation is only available in optimal control problems; there is no similar formulation in optimal design of systems subjected to static loads.

In terms of the discretization techniques, several different ways can be used: direct discretization by finite differences, piece-wise polynomials and splines of various orders. Table 2 lists the advantages and disadvantages of different discretization techniques. For the piece-wise polynomial and spline interpolations, the number of variables is usually not very large; therefore, the resulting NLP is not too large. Sparsity can be utilized in the formulations, but not necessary. These formulations provide good smoothness for the final solution; therefore, they are well suitable for digital human motion simulation. However, the drawback of these formulations is that the implementation is not straightforward. They are sometimes too restrictive; therefore no solution or no good solution may be obtained for the optimal control problem. The finite difference method usually requires large number of grid points N; therefore large-scale NLP algorithms with sparse matrix capabilities are required to solve the problems efficiently.

Table 1. Advantages and disadvantages of different formulations

Formulations	Variables	Advantages	Disadvantages
State variable elimination	$\boldsymbol{\tau}$	1. Small optimization problems. 2. Equations of motion are satisfied at each iteration; intermediate solutions may be usable. 3. Error in the solution of DEs can be controlled.	1. Equations of motion must be integrated at each iteration, which is expensive. 2. A good DEs integrator is needed. 3. Objective or constraints involving $\mathbf{q}$ are implicit functions of the variables; their evaluation requires solution of the equations of motion. 4. Design sensitivity analysis must be performed, and its implementation is tedious. 5. Dense constraint Jacobian and Hessian matrices; difficult to treat large-scale problems.
Direct collocation	$\mathbf{q}, \boldsymbol{\tau}$	1. Formulations are explicit in terms of variables. 2. Equations of motion are not integrated at each iteration. 3. Constraint Jacobians and Hessian are sparse. 4. Design sensitivity analysis is not needed. 5. Constraints on $\mathbf{q}$ and $\boldsymbol{\tau}$ can be treated efficiently.	1. Numbers of variables and constraints are large. 2. Intermediate solutions may not be usable. 3. Optimization algorithms for large-scale problems must be used. 4. For efficiency, advantage of sparsity of the constraint Jacobians and Hessians must be utilized. 5. Optimization variables sometimes need to be normalized.
Differential inclusion	$\mathbf{q}$	1. Smaller number of optimization variables. 2. Formulations are explicit in terms of variables. 3. Equations of motion are not integrated, but are satisfied at each iteration. 4. Intermediate solutions may be usable. 5. Design sensitivity analysis is not needed. 6. Constraints on $\mathbf{q}$ can be treated efficiently.	1. Objective or constraints involving $\boldsymbol{\tau}$ need evaluation of inverse dynamics. 2. Implementation sometimes is not straightforward.

Table 2. Advantages and disadvantages of different discretization techniques

Formulation	Advantages	Disadvantages
Finite difference	• Very sparse constraint Jacobians and Hessian. • Implementation is very straightforward. • State or control variable constraints become simple bounds in most cases.	• Larger number of variables and constraints. • Optimization algorithms for large-scale sparse problems must be used.
Piece-wise polynomial interpolation	• Good smoothness. • Smaller NLP problems. • State or control variable constraints become simple bounds or linear in most cases	• The required curve profile may be too restrictive. • Implementation sometimes is not straightforward.

Table 3. Numbers of non-zero elements in gradient and Jacobian for different formulations

Item		Formulation		
		CD	**Hermite**	**B-spline**
Gradient vector	**Objective function (13)**	$d(N+3)$	$2d(N+2)$	dn
Jacobian of constraints	**Foot ground penetration constraints**	$3dN_p(N+1)$ $[dN_p(N+3)(N+1)]$*	$4dN_p(N+1)$ $[2dN_p(N+2)(N+1)]$	$4dN_p(N+1)$ $[dnN_p(N+1)]$
	Foot strike position constraints	$3dN_s(N+1)$ $[dN_s(N+3)(N+1)]$	$4dN_s(N+1)$ $[2dN_s(N+2)(N+1)]$	$4dN_s(N+1)$ $[dnN_s(N+1)]$
	ZMP stability constraints	$3dN_Z(N+1)$ $[dN_Z(N+3)(N+1)]$	$4dN_Z(N+1)$ $[2dN_Z(N+2)(N+1)]$	$4dN_Z(N+1)$ $[dnN_Z(N+1)]$
	Joint angle constraints	-	-	$4d(N+1)$ $[d^2n(N+1)]$
Total of 1st order derivatives (Gradient vector & Jacobian)		$3d(N_p+N_s+N_Z)(N+1)+$ $d(N+3)$ $[d(N+3)\cdot$ $((N_p+N_s+N_Z)(N+1)+1)]$	$4d(N_p+N_s+N_Z)(N+1)+$ $2d(N+2)$ $[2d(N+2)\cdot$ $((N_p+N_s+N_Z)(N+1)+1)]$	$4d(N_p+N_s+N_Z+1)(N+1)+$ dn $[dn\cdot$ $((N_p+N_s+N_Z+d)(N+1)+1)]$

* The expressions in the brackets give the total number of elements when sparsity is not considered.

Table 3 shows the approximated numbers of non-zero elements in the gradient vector and Jacobian of constraint functions for all the formulations. The following symbols are used in the table: d = number of degree of freedoms (DOFs) in the human model; N = number of time intervals (number of time grid points = $N+1$); n = number of control points in a cubic B-spline; N_g = number of foot ground penetration constraints; N_s = number of foot strike position constraints; N_Z = number of ZMP stability constraints. This approximation in fact provides upper bounds for the numbers of non-zero elements in the vectors and matrices. Only the storage of the input information for the algorithm, such as the Jacobian matrix, is discussed and compared.

VI. A Numerical Example

All the formulations developed in Section V are applied to a gait simulation example for evaluation. All the formulations are solved using the sparse SQP algorithm in SNOPT[34]. A PC with 3.2 GHz processor and 1.0 GB RAM is used for running the programs and recording the relative CPU times. Each solution case of the example problem was run several times with different starting point and the shortest time was recorded. Results of the examples are listed and compared. Advantages and disadvantages of the formulations are discussed.

A. A 18-DOF Human Lower-Body Gait Model

An 18-DOF three-dimensional digital human lower-body model is considered as shown in Figure 4[18]. In this model, the hip has 6 global DOFs, including 3 translations and 3 rotations. The pelvis has 3 rotational DOFs, and the knee has 1 rotational DOF. The ankle is represented by 2 orthogonal rotational joints. The two legs are exactly symmetric. The number of foot ground penetration constraints in this example is 8, which is the total number of points on the two feet. One foot strike position constraint and four ZMP stability constraints are used in this study. Note also that since the ground reaction forces are not considered in the current formulation, the torques are only due to gravity, inertia, and Coriolis effects.

The total motion time is considered as 2.442 seconds. No special techniques are used to find an initial point for the formulations. The starting values for the optimization variables are taken as zero. Table 4 gives the numbers of iterations and CPU (s) for different formulations.

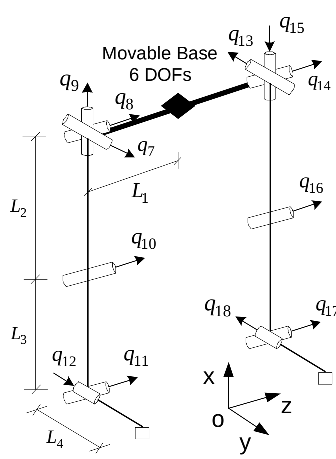


Figure 4. A 18-DOF human lower-body gait model

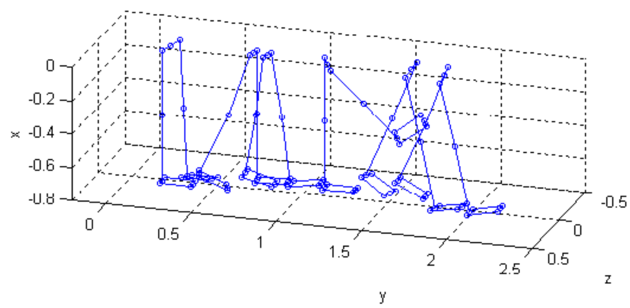
B. Discussion of Results

1. Number of time steps

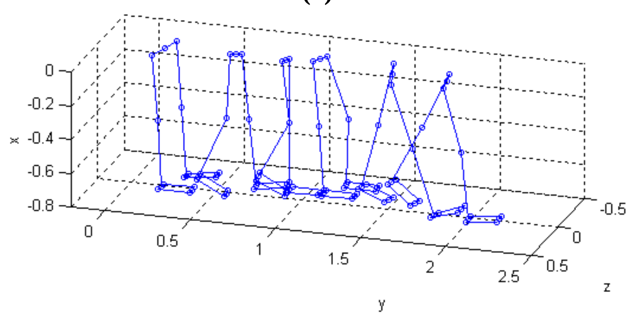
It is obvious that the number of time steps used in the numerical solution process can affect the performance of the formulations. In general if the step size is large, the size of the optimization problems is small. The numbers of iterations and CPU times to find an optimal solution are small, and vise versa. If the step size is too small, the sizes of the alternative formulations become very large which requires additional calculations and computer storage. To evaluate the performance of various formulations, a few different grid sizes are tried for the example. All the three formulations are indeed very sparse, with small numbers of non-zero elements in the constraint Jacobian. Table 4 shows that as the number of time steps is increased the computational effort with all the formulations also increases. It is also observed that the CD and B-spline formulations require less computational efforts compared to the formulation based on Hermite interpolation. For smaller numbers of time steps, the CD formulation is very efficient. When the number of time steps becomes larger, CD and B-spline formulations require similar computational efforts.

Table 4. Numbers of iterations and CPU (s) for different formulations

No. of Time Intervals (N)	Numbers of Iterations			CPU (s)		
	CD	Hermite	B-spline	CD	Hermite	B-spline
12	30	31	43	11	20	28
36	29	32	37	65	328	246
72	96	49	51	1398	2047	1503



(a)



(b)

Figure 5. Gait motions of a 18-DOF human lower-body model (a) kinematically feasible gait where torques are not considered; (b) torques are considered

2. Gait simulations

Figure 5 show the lower-body gait simulations. For simplicity of illustration, the axis in the walking direction (y) is extended. It is seen that all the constraints presented in Section III, including the foot ground penetration and prescribed strike locations are all satisfied. Figure (5a) shows a kinematically feasible walking motion, while (5b) illustrates an optimal gait motion with the absolute torque values are minimized. Note that since the joint torques are not considered in (5a), some motions such as the high kick of the right leg in Fig. (5a) are not realistic. Figure (5b) shows a more realistic gait motion.

3. Advantages and disadvantages of formulations

The main advantages of the differential inclusion inverse-dynamics based formulations for dynamic systems are summarized as follows: (i) the equations of motion for the system need not be integrated explicitly; therefore, unnecessary simulations of the system are avoided at intermediate designs; (ii) design sensitivity analysis of the systems is not needed since all the problem functions are explicit in terms of the variables. The major disadvantage of the formulation based on B-spline interpolation is that some constraints are linear instead of simple bounds, such as the joint angle constraints.

4. Other formulations

It is clear that other simultaneous formulations are possible. These are based on different discretization techniques of the first order or second order EOMs[29], and piecewise higher degree polynomial approximations of state variables[27]. However, these multi-step methods or higher degree of polynomials may result in significant complexity of numerical implementation for the formulations, which is not desired. The application of these methods for digital human motion prediction needs further evaluation.

5. Future research

Optimization-based gait prediction reveals great insights into real human gait. Intuitively, simulation of a natural gait will involve multiple objectives. This is an important topic for future consideration in simulation of human gait. A full-size three-dimensional gait model including human upper body and arm movements will be very useful for understanding of upper body motion. Human running and jumping motions are of special interest to certain types of applications, such as sports. Some further research in human motion prediction and simulation also includes comparison with other models now being used, and the consideration of collision avoidance in the formulations. A general framework for collision avoidance of human motions has been recently presented[35].

VII. Concluding Remarks

The task-based motion prediction of digital humans was shown to be an optimal control problem and therefore could be solved numerically by powerful sparse NLP techniques. In order to provide clues on how best to formulate the problem for practical applications, various solution techniques for optimization-based optimal control were presented and compared. Different discretization techniques were discussed and their strength and weakness compared. Based on this work, the following observations are made.

1. The direct collocation and differential inclusion methods did not require integration of the equations of motion. All functions of the formulations became explicit in terms of the optimization variables
2. Differential inclusion formulation is unique in the sense that it is only available in optimal control problems. Compared to direct collocation method, differential inclusion formulation has special advantage and great potential for large-scale digital human motion prediction, because the size of the optimization problem is smaller.
3. Formulations based on central difference and B-spline required less computational efforts compared to the formulation based on Hermite interpolation for the same number of time steps.
4. When the number of time steps becomes larger, CD and B-spline formulations require similar computational efforts. The differential inclusion formulation based on B-spline is currently being implemented in the virtual human environment *SANTOS*^TM[35].

More work is still needed to fully develop nonlinear optimization-based control techniques, such as differential inclusion for realistic human motion prediction, with the validation of motion tracking data. More justification should be provided for the selection of these methods for human motion simulation. The current control torque does not include ground reaction forces, so the torque in the stance leg is not real; the ground reaction forces need to be included in future work.

Acknowledgement

Support for this research provided by the Virtual Soldier Research program at the University of Iowa, is gratefully acknowledged.

References

[1]Furusho, J. and Masubuchi, M. “Control of a dynamical biped locomotion system for steady walking,” *Journal of Dynamic System, Measurement, and Control*. Vol. 108, 1986, pp. 111-118.

[2]Faraway, J.J., Zhang, X. and Chaffin, D.B., “Rectifying postures reconstructed from joint angles to meet constraints,” *Journal of Biomechanics*. Vol. 32 (7), 1999, pp. 7333-736.

[3]Yamaguchi, J., Soga, E., Inoue, S. and Takanishi, A., “Development of a bipdal humanoid robot control method of whole body cooperative dynamic biped walking,” *IEEE International Conference on Robotics and Automation*, Vol. 1, 1999, pp. 368-374.

[4]Nishiwaki, K. and Kagami, S., “Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired ZMP,” *IEEE International Conference on Intelligent Robots and Systems*, Vol. 3, 2002, pp. 2684-2689.

[5]Park, J. and Kim, K., “Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control,” *IEEE International Conference on Robotics and Automation*, Vol. 4, 1998, pp. 3528-3533.

[6]Chevallereau, C. and Aoustin, Y. “Optimal reference trajectories for walking and running of a biped robot,” *Robotica*, Vol. 19, 2001, pp. 557-569.

[7]Bessonnet, G., Sardain, P. and Chesse, S., “Optimal motion synthesis- dynamic modeling and numerical solving aspects,” *Multibody System Dynamics*, Vol. 8(3), 2002, pp. 257-278.

[8]Saidouni, T. and Bessonnet, G., “Generating globally optimized sagittal gait cycles of a biped robot,” *Robotica*, Vol. 21(2), 2003, pp. 199-210.

[9]Bessonnet, G., Seguin, P. and Sardain, P., “A parametric optimization approach to walking pattern synthesis,” *The International Journal of Robotics Research*, Vol. 24(7), 2005, pp. 523-536.
[10]Lo, J.; Huang, G. and Metaxas, D., “Human motion planning based on recursive dynamics and optimal control techniques,” *Multibody Sys. Dyn.* Vol. 8(4), 2002, pp. 433-458.
[11]Yamaguchi, G.T. and Zajac, F.E., “Restoring unassisted natural gait to paraplegics via functional neuromuscular stimulation: a computer simulation study,” *IEEE Trans. Biomed. Eng.*, Vol. 37, 1990, pp. 886–902.
[12]Pandy, M.G., Anderson, F.C. and Hull, D.G., “A Parameter optimization approach for the optimal control of large-scale musculoskeletal systems,” *J. Biomechanical Eng., Transactions of the ASME*, Vol. 114(4), 1992, pp. 450-460.
[13]Anderson, F.C. and Pandy, M.G., “Dynamic optimization of human walking,” *Journal of Biomechanical Engineering*, Vol. 123(5), 2001, pp. 381-390.
[14]Thelen, D.G., Anderson, F.C. and Delp, S.L., “Generating dynamic simulations of movement using computed muscle control,” *Journal of Biomechanics*. Vol. 36, 2003, pp. 321-328.
[15]Denavit, J. and Hartenberg, R.S., “A Kinematic notation for lower-pair mechanisms based on matrices,” *J. Applied Mechanics*, Vol. 77, 1955, pp. 215-221.
[16]Hollerbach, J.M., “A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity,” *IEEE Transactions on Systems, Man, and Cybernetics*. SMC-10 (11), 1980, pp. 730-736.
[17]Toogood, R.W., “Efficient robot inverse and direct dynamics algorithms using micro-computer based symbolic generation,” *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 3, 1989, pp. 1827-1832.
[18]Xiang, Y.-J., Chung, H.-J., Arora, J.S., and Abdel-Malek, K., *Digital Human Modeling and Virtual Reality for FCS - Simulation of Human Gait*, Technical Report No. VSR-06.01, CCAD, University of Iowa, Iowa City, IA 52242.
[19]Wang, Q., Xiang, Y.-J., Kim, H.-J., Arora, J.S. and Abdel-Malek, K. “Alternative formulations for optimization-based digital human motion prediction,” Paper 2005-01-2691, *2005 Digital Human Modeling for Design and Engineering Symposium*, Iowa City, IA, June 14-16, 2005.
[20]Yang, J., Marler, R.T., Kim, H., Arora, J. and Abdel-Malek, K., “Multi-objective optimization for upper body posture prediction,” *Proc. 10th AIAA/ISSMO Multi-disciplinary Analysis and Optimization Conf.*, August, Albany, NY, 2004.
[21]Arora, J.S., “Optimization of structures subjected to dynamic loads,” in *Structural Dynamic Systems Computational Techniques and Optimization: Optimization Techniques*, edited by Leondes, C.T., Gordon and Breach Science Publishers, 1999, pp. 1-72.
[22]Goswami, A., “Postural stability of biped robots and the foot rotation indicator point,” *International Journal of Robotics Research*, Vol. 18(6), 1999, pp. 523-533.
[23]Vukobratović, M. and Borovac, B., “Zero-moment point – thirty five years of its life,” *International Journal of Humanoid Robotics*, Vol. 1(1), 2004, pp. 157-173.
[24]Roussel, L., Canudas-de-Wit, C. and Goswami, A., “Generation of energy optimal complete gait cycles for biped robots,” *Proc. of IEEE International Conference on Robotics and Automation*. (held in Leuven, Belgium), Vol. 3, 2001, pp. 2036-2041.
[25]Neptune, R.R. and Hull, M.L., “Evaluation of performance criteria for simulation of submaximal steady-state cycling using a forward dynamic model,” *J. Biomechanical Eng.* Vol. 120, 1998, pp. 334-340.
[26]Biegler, L.T., Ghattas, O., Heinkenschloss, M., Bloemen Waanders, B.v. (Eds.), *Large-Scale PDE-Constrained Optimization, Lecture Notes in Computational Science and Engineering*, Vol. 30, Springer-Verlag, Berlin, 2003.
[27]Herman, A.L. and Conway, B.A., “Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules,” *J. Guidance, Cont. Dyn.* Vol. 19(3), 1995, pp. 592-599.
[28]von Stryk, O. and Bulirsch, R., “Direct and indirect methods for trajectory optimization,” *Annals of Operations Research*, Vol. 37, 1992, pp. 357-373.
[29]Betts, J.T., “Survey of numerical methods for trajectory optimization,” *J. Guidance, Cont. Dyn.* Vol. 21(2), 1998, pp 193-207.
[30]Schulz, V., “Simultaneous solution approaches for large optimization problems,” *J. Comp. Appl. Math.* Vol. 164(1), 2004, pp. 629-641.
[31]Hull, D.G., *Optimal Control Theory for Applications*, Springer-Verlag, New York, 2003.
[32]Kaplan, M.L. and Heegaard J.H., “Predictive algorithms for neuromuscular control of human locomotion,” *J. Biomechanics*, Vol. 34(8), 2001, pp. 1077-1083.
[33]Piegl, L. and Tiller, W., *The NURBS Book, 2nd Edition*, Springer-Verlag, New York, NY, 1997.
[34]Gill, P.E., Murray, W. and Saunders, M.A., “SNOPT: An SQP algorithm for large-scale constrained optimization,” *SIAM J. Optim.* Vol. 12, 2002, pp. 979-1006.
[35]Yang, J., Marler, T., Beck, S., Kim, J., Wang, Q., Zhou, X., Pena Pitarch, E., Farrell, K., Patrick, A., Potratz, J., Abdel-Malek, K., Arora, J.S. and Nebel, K., “New capabilities for the virtual-human SANTOS™,” *SAE 2006 World Congress*, Detroit, MI, April 3-6, 2006.