Alternative formulations for optimization-based digital human motion prediction

Q. Wang, Y.-J. Xiang, H.-J. Kim, J. S. Arora, and K. Abdel-Malek
Virtual Soldier Research Program (VSR), The University of Iowa, Iowa City, IA 52242, USA

ABSTRACT

Simulating human motion is a complex problem due to redundancy of the human musculoskeletal system. The concept of task-based motion prediction using single- or multi-objective optimization techniques provides a viable approach for predicting intermediate motions of digital humans. It is shown that task-based motion prediction is in fact a numerical optimal control problem. Alternative formulations for simulation of human motion are possible and can be solved by modern nonlinear optimization methods. Three techniques based on state variable elimination, direct collocation and differential inclusion are presented and compared. The basic idea of the formulations is to treat different combinations of the state variables, such as the joint profiles and torques or their parametric representations as independent variables in the optimization process. Different ways to discretize the equations of motion are also presented, namely finite difference, piecewise polynomial interpolation and series expansion. The advantages and disadvantages of different formulations are discussed. A numerical example is used to illustrate the basic ideas, and is solved by a large-scale sparse nonlinear programming solver. It is concluded that with the aid of motion tracking for validation, optimal control techniques based on nonlinear optimization have great potential to provide a useful tool for realistic human motion prediction.

INTRODUCTION

Simulating human postures and intermediate motion is traditionally a complex and difficult problem due to redundancy of the human musculoskeletal system and other biomechanical factors (Jung and Choe 1996; Chaffin 1997; Hase and Yamazaki 1997; Faraway *et al*. 1999; Abdel-Malek *et al*. 2002). The concept of task-based posture and motion prediction and multi-objective optimization techniques provide a viable approach for predicting natural posture and intermediate motions of digital humans represented by a relatively large number of degrees of freedom. Various approaches and numerical algorithms for implementing task-based posture prediction using multi-objective optimization algorithms have been studied by Mi (2004), Marler (2004), and Yang *et al*. (2004). Although success has been achieved, the application is limited to kinematics.

Realistic virtual human modeling and simulation and natural-looking motion require consideration of several topics when external loads need to be carried, such as muscle forces, muscle fatigue and anthropometric data. Therefore, a formulation for prediction of the motion of digital humans that includes all these effects and a solution procedure are needed. Basically, in a specified time range, equations of motion (dynamic equilibrium equations) and constraints need to be satisfied for the system. For realistic human motion prediction, the objectives are various human performance indexes, such as discomfort, potential energy, or a combination of them. The resulting optimization problem is therefore shown to be an optimal control problem (OCP) (Bryson and Ho 1975). Optimal control techniques have long been applied to various human models (Chow and Jacobson 1971; Audu and Davy 1988; Khang and Zajac 1989a,b; Pandy *et al*. 1991, 1992, 1995; Anderson *et al*. 1995; Tashmann *et al*. 1995). However most numerical techniques treat only the control (force) variables as unknowns; the state variables are obtained by integration of the dynamic equations of motion (Pandy *et al*. 1992; Neptune and Hull 1998). Recently, the direct collocation concept has been applied to large-scale neuromuscular control of human motion (Kaplan and Heegaard 2001, 2002).

It is well known that OCP can be solved efficiently by nonlinear optimization techniques (Kraft 1985; Betts 1998; Hull 1997, 2003). The basic idea is to discretize the system of differential equations, and define finite dimensional approximation or parametric representation for the state and control variables, converting the optimal control problem into a nonlinear programming problem (NLP), which is solved numerically. Several viable formulations are available. If only the state variables are treated as optimization variables, while the control variables are not included in the formulation, this is called *state variable elimination*, or *direct shooting* (Tseng and Arora 1989; Arora 1999). When the state and control variables are all treated as optimization variables, the approach is called the *direct collocation/transcription* method (Hargraves and Paris 1987; Betts 1998). If the control variables are eliminated from the system, i.e., only the state variables are treated as optimization variables, the approach is called the *differential inclusion* method (Seywald 1994; Coverstone-Carroll and Williams 1994; Kumar and Seywald 1996). Conway and Larson (1998) presented a comparison of collocation and differential inclusion methods in direct trajectory optimization. Another possibility is the so-called *multiple shooting* technique (Leineweber *et al.* 2003).

For an intermediate digital human model, the Denavit-Hartenberg (DH) method is used for modeling human biomechanics and dynamics. The differential equations are discretized and finite dimensional approximation or parametric representation for the joint angles and torque variables are defined, converting the simulation problem into a NLP problem. Different discretization techniques for the equations of motion are available. These techniques include finite difference, piecewise polynomial interpolation and series expansion. In some formulations, explicit integration of the equations of motion is avoided, which is very advantageous for large-scale problems. With these formulations, all the design constraints, i.e., limits on joint angles, velocities, accelerations, and joint torques can be expressed explicitly in terms of the optimization variables. Therefore their gradient evaluations become simple. Although these formulations are well studied for aerospace and chemical engineering control problems, their application to digital human simulation is relatively new. Limited research can be found for the application of direct collocation and differential inclusion to optimization based human simulation. However, with powerful NLP solvers, these methods show great potential for digital human motion prediction. A comprehensive presentation of the alternative formulations for digital human motion prediction based on numerical optimal control techniques is given in this paper. These formulations may offer some clues on how best to formulate the problem for practical applications. Numerical examples are optimized and their solutions are compared with those available in the literature. Advantages and disadvantages of the formulations are also discussed. Nonlinear optimization-based control techniques, such as direct collocation and differential inclusion show potential for realistic human motion prediction. However, further research is needed to fully apply the numerical techniques for large-scale human models, when combined with experimental data from motion tracking tools.

HUMAN MOTION MODELING

Human limbs and joints can be modeled 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.

DH PARAMETERS AND TRANSFORMATION MATRICES

The DH method is an approach for relating the position of a point in one coordinate system to another one, by using transformation matrices (Denavit and Hartenberg 1955; Paul 1981). In order to obtain a systematic representation of a serial kinematic chain (Figure 1), $\mathbf{q} \in R^n$ is defined as the vector of *n*-generalized coordinates. The position vector function of a point of interest on the end-effector of a human model in the Cartesian space can be written in terms of the joint variables as

$$\mathbf{X} = \mathbf{X}(\mathbf{q}) \tag{1}$$

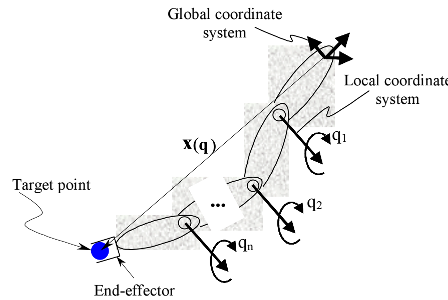

Figure 1. General kinematic system

where $\mathbf{X}(\mathbf{q})$ can be obtained from the multiplication of the 4×4 homogeneous transformation matrices ${}^{i-1}\mathbf{T}_i$ defined by the DH method such that

$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

in which ${}^{i-1}\mathbf{T}_i$ is the rotation matrix relating coordinate frames *i* and *j*, representing by four parameters θ_i, d_i, α_i, and a_i, as shown in Figure 2.

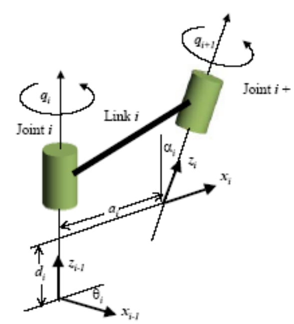

Figure 2. Joint coordinate systems

Use augmented 4×1 vectors ${}^0\mathbf{r}_n$ and $\mathbf{r}_n$ to express Cartesian vectors $\mathbf{X}(\mathbf{q})$ and $\mathbf{X}_n$ characterizing the set of points touched by the end-effector as:

$${}^0\mathbf{r}_n = \begin{bmatrix} \mathbf{X}(\mathbf{q}) \\ 1 \end{bmatrix}, \quad \mathbf{r}_n = \begin{bmatrix} \mathbf{X}_n \\ 1 \end{bmatrix} \tag{3}$$

where $\mathbf{X}_n$ is the position of the end-effector with respect to the nth coordinate system. Using these relationships, ${}^0\mathbf{r}_n$ can be written as

$${}^0\mathbf{r}_n = {}^0\mathbf{T}_n(\mathbf{q})\mathbf{r}_n \tag{4}$$

where

$$ {}^{0}\mathbf{T}_n(\mathbf{q}) = \prod_{i=1}^{n} {}^{i-1}\mathbf{T}_i = {}^{0}\mathbf{T}_1(q_1)\,{}^{1}\mathbf{T}_2(q_2)\cdots{}^{n-1}\mathbf{T}_n(q_n) \qquad (5) $$

GOVERNING EQUATIONS

The general form of dynamic equations of motion is derived from energy principle. Assuming that only the gravity forces and the driving joint torques are applied to the human links, the Lagrange's equation is written as follows:

$$ \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i, \quad i = 1,..,n \qquad (6) $$

where L is the *Lagrangian*; q_i is the generalized coordinate of joint i; τ_i is the generalized torque of joint i actuated by human muscles; n is the number of the total degrees of freedom (DOF), and t is time, respectively.

The general equation of motion including several external loads can be derived as (Kim 2004)

$$ \boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) + \sum_j \mathbf{J}_j^T m_j \mathbf{g} + \sum_k \mathbf{J}_k^T \mathbf{F}_k + \mathbf{K}\left(\mathbf{q} - \mathbf{q}^N\right) \qquad (7) $$

where $\mathbf{q}$ $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ are the joint angle, velocity and acceleration vectors. $\mathbf{M}(\mathbf{q})$ is the mass-inertia symmetric matrix, and $\mathbf{V}(\mathbf{q},\dot{\mathbf{q}})$ is the Coriolis and Centrifugal vector, respectively. The third, fourth and last terms of the right hand side of Eq. (7) represent joint torque vectors due to gravity forces, external forces, and muscle elasticity. In a simplified way of presentation, Eq. (7) is

$$ \boldsymbol{\tau} = \mathbf{f}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},t) \qquad (8) $$

Note that $\mathbf{q}$ $(n \times 1)$ represents the state variables for the problem. The equation of motion can be written in a first-order form, the so-called *state space* representation, as:

$$ \dot{\mathbf{Q}}(t) = \mathbf{F}(\mathbf{Q}(t), \boldsymbol{\tau}, t) \qquad (9) $$

where $\mathbf{Q}^T = [\mathbf{q} \quad \dot{\mathbf{q}}]$ $(2n \times 1)$. It is worthwhile sometimes to examine both the first order form, and the second order form of the equations as well. Although a large number of accurate, high-order or multi-step numerical methods are available to solve the system of first order differential equations (DEs) (Atkinson 1988). Numerical methods are available to solve the second order DEs directly (Bathe 1982). It is noted that in the equations of motion (8) or (9), the joint profiles and torques are all unknowns; therefore, it is an indeterminate system and is not directly solvable by forward dynamics or inverse dynamics.

HUMAN MOTION PREDICTION AS AN OPTIMIZATION PROBLEM

OBJECTIVE FUNCTION

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

$$ J = c_0(\mathbf{q}(T),T) + \int_0^T h_0(\boldsymbol{\tau},\mathbf{q},t)\,dt \qquad (10) $$

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, 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 simulation, the objective functions represent human performance measures. Various performance measures of digital human have been developed, such as the *joint displacement* and *effort*, which can be essentially expressed as (Yang *et al*. 2004; Marler 2004)

$$ J = \sum_{i=1}^{n} w_i \left(q_i - q_i^N\right)^2 \qquad (11) $$

where w_i is the weight coefficient, and q_i^N is a natural or initial joint angle for the ith DOF. Other performance measures include *delta-potential-energy*, as (Marler 2004),

$$ J = \sum_{i=1}^{K} (m_i g)^2 (\Delta h_i)^2 \qquad (12) $$

where $(m_i g)^2$ represents the weights and $(\Delta h_i)^2$ is the vertical distance of lumped masses, respectively. K is the total number of lumped masses. Marler (2004) also considered *discomfort* as one cost function,

$$ J = \frac{1}{G}\sum_{i=1}^{n}\left[w_i\left(\Delta q_i^{norm}\right) + G \times QU_i + G \times QL_i\right] \qquad (13) $$

in which Δq_i^{norm} is the normalized joint variable and G is a weighting term. QU_i and QL_i are functions of the lower and upper limits for the ith joint variable, representing penalties when a joint is near its limits. Muscle energy consumption functions have been reported by Alexander (1997) and Kim (2004), and some other functions by Mi (2004).

For multiobjective performance measures, different combinations of objective functions can be used (Arora 2004; Marler and Arora 2004). A weighted sum might be an easy choice, which entails minimizing the following aggregate function:

$$ \overline{J} = \sum_{i=1}^{nc} \gamma_i J_i \qquad (14) $$

where nc is the total number of objective functions considered. γ_i represents relative importance of each objective, and it is dependent upon users' preference. This method always yields a Pareto optimal point (Marler and Arora 2004).

DESIGN CONSTRAINTS

The design constraints consist the equations of motion in Eq. (8) or (9), and other design requirements as functional constraints, as

$$ \mathbf{g}(\boldsymbol{\tau},\mathbf{q}) = \mathbf{c}(\mathbf{q}(T),T) + \int_0^T \mathbf{h}(\boldsymbol{\tau},\mathbf{q},t)\,dt \le \mathbf{0} \qquad (15) $$

Note that these 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, as

$$ \mathbf{q}(0) = \mathbf{q}_0, \quad \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0 \qquad (16) $$

$$ \boldsymbol{\varphi}(\mathbf{q}(T),T) = \mathbf{0} \qquad (17) $$

The other 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 $t \in [0,T]$

$$\mathbf{g}(\boldsymbol{\tau}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) \leq \mathbf{0} \tag{18}$$

Equality constraints if present in the formulation can be treated quite routinely. Five treatments of the point-wise constraints in Eq. (18) have been discussed in the literature (Tseng and Arora 1989; Arora 1999). In dynamic systems, the time-dependent inequality constraints in Eq. (18) may include the joint angle, velocity and acceleration constraints, end-effectors, etc., to ensure that an appropriate human motion is predicted.

OPTIMIZATION FORMULATION

Given the above-mentioned variables, constraints, and human performance measures, the optimization based motion prediction is determined by solving the following problem:

Find: $\mathbf{q}, \boldsymbol{\tau} \in R^n$ to minimize the human performance objective in Eq. (14), subject to the equations of motion (8) or (9), end-effector, and the constraints on the joint profiles and torques, as

$$\mathbf{q}^L \leq \mathbf{q} \leq \mathbf{q}^U \tag{19}$$

$$\boldsymbol{\tau}^L \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}^U \tag{20}$$

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. Other constraints can be added similarly. Analytical gradients can be calculated for all functions according to different solution techniques. This problem can be solved by the Pontryagin method (Pontryagin *et al*. 1962), or by numerical optimization techniques discussed later in the paper. 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.

TIME AS AN OPTIMIZATION VARIABLE

In simulating digital humans, the total time T can be specified for a motion in Eqs. (8) or (9). However, sometimes it is desirable to consider time as an objective or constraint, such as the simulation of a motion within certain time limits. In these cases, total time T can be included as one more optimization variable. The minimum time problem has the objective function as

$$\text{Minimize} \quad T \tag{21}$$

The problem of limiting time to a certain range involves the following time constraint

$$T^L \leq T \leq T^U \tag{22}$$

where T^L and T^U are the lower and upper bounds on the total time T.

SOLUTION TECHNIQUES OF THE OPTIMAL CONTROL PROBLEM

Several different ways based on numerical optimization to solve the OCP problem are available, namely state variable elimination, direct collocation, and differential inclusion. The details of these formulations for motion prediction are listed in Table 1.

STATE VARIABLE ELIMINATION

This is also called direct shooting method. In this formulation, the joint profiles are eliminated form the optimization problems; only the joint torques $\boldsymbol{\tau}$ are treated as unknowns and optimized. The joint profiles, such as the joint angle, velocities and accelerations are obtained through numerical integration of the equations of motion (8) or (9) in each design cycle. Note that $\mathbf{f}^{-1}$ denotes numerical integration in Table 1. Design sensitivity analysis is needed in this formulation, since the joint angle, velocities and accelerations are implicit functions of the optimization variables, the joint torques. Finite difference, *direct differentiation* method and *adjoint variable* method are available to calculate the derivative information for the objective and constraint functions (Pandy *et al*. 1992; Hsieh, and Arora 1984; Arora 1999). All these methods involve numerical integration, which makes the formulation computationally very expensive.

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 (8) or (9) are treated as equality constraints in the formulation. The more general forms of direct collocation is the so-called *partial differential equation (PDE) constrained optimization* (Biegler *et al*. 2003), and *mathematical programming with equilibrium constraints* (*MPEC*) (Luo *et al*. 1996). The direct collocation approach has long been used in most engineering fields, such as the *simultaneous analysis and design* of FEM-based structural optimization (Haftka 1985), open-loop optimal control problems for trajectory design in aerospace engineering (Hargraves and Paris 1987; Enright and Conway 1991; Betts and Huffman 1992; Betts 2000), dynamic chemical process engineering (Biegler 1988; Cervantes and Biegler 1998), and Robotic or human motion planning (Kaplan and Heegaard 2001, 2002; Lo *et al*. 2002; Bottasso and Croce 2004). Different optimization algorithms have been used for direct collocation or multiple shooting, among which the sequential quadratic programming (SQP) (Hargraves and Paris 1987, Tanartkit and Biegler 1996, Betts 2000; Schulz 2004), and interior point algorithm (Cervantes *et al.* 2000) are popular choices. Since the resulting NLP is large and sparse, sparse NLP were extensively discussed (Betts and Huffman 1992; Cervantes and Biegler 1998).

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 formulations 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 are exploited in the optimization process. The resulting optimization problem is solved using powerful sparse nonlinear programming algorithms, such as SQP.

DIFFERENTIAL INCLUSION

If the joint torques are eliminated form the optimization problems so that the unknowns are the joint profiles, such as the discretized joint angle, velocities and accelerations, or parametric representations of the joint profiles, this is the differential inclusion formulation. Note that in this formulation, the governing differential equations are not integrated nor treated as equality constraints; instead an inverse dynamics procedure can be used to calculate the joint torques, based on the current joint profiles. The equations of motion (8) or (9) are automatically satisfied in the optimization process.

DISCRETIZATION TECHNIQUES OF THE EQUATIONS OF MOTION

Different discretization techniques for the equations of motion (8) or (9) have been studied in the literature. The general idea is to transfer the DEs to an algebraic system of equations, the so-called defect equations, which need to be set to zero to enforce the DEs. In general there are three classes of methods. One is finite difference (Wang and Arora 2004). One is to use some polynomial interpolation between the time grid points (Betts 1998). The last one is to use a series expansion in terms of orthogonal polynomials, such as Legendre or Chebyshev polynomials (Vlassenbroeck 1988; Fahroo and Ross 2002). A relative comparison of the three techniques is listed in Table 3. For the second method, the most common ways are the trapezoidal or Simpson's quadrature schemes based on piecewise quadratic or cubic polynomials (Hargraves and Paris 1987; Betts 1990; Enright and Conway 1992; von Stryk and Bulirsch 1992). Explicit or implicit Runge-Kutta methods were used to discretize the state equations by Albuquerque and Biegler (1997), Biehn *et al*. (2000), and Betts *et al*. (2000, 2002). Higher degree polynomials for direct collocation were studied by Herman and Conway (1995), and Hu *et al*. (2002). If very smooth trajectory is required, B-spline curves can also be used to parameterize the dynamic equations (Chen 1991; Wang *et al*. 2001; Lo *et al*. 2002).

If the equations of motion in (8) or (9) are directly discretized into $N+1$ equations in the entire time interval $[0,T]$, the Eqs. (8) or (9), and the inequality constraints in Eq. (18) represent in fact the constraints that need to be satisfied at each time grid point, as

$$\boldsymbol{\tau}_i = \mathbf{f}(\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i, t_i), \quad i = 0,..,N \tag{23}$$

$$\mathbf{g}_i(\boldsymbol{\tau}_i, \mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i, t_i) \le \mathbf{0}, \quad i = 0,..,N \tag{24}$$

The length of the subinterval is

$$\Delta t = \frac{T}{N} \tag{25}$$

If the parametric representations of the state variables are used as variables, such as the coefficients of the series expansion, and control points for B-splines, the equations of motion in (8) or (9) are reduced to a system of algebraic equations, as

$$\mathbf{H}(\mathbf{P}) = \mathbf{0} \tag{26}$$

and Eq. (18) is rewritten as

$$\mathbf{g}(\mathbf{P}) \le \mathbf{0} \tag{27}$$

where vector $\mathbf{P}$ contains the unknown parameters or control points used to represent the joint profiles, which will be treated as variables in the optimization process.

FINITE DIFFERENCE

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 (CD), 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:

$$\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_i(t_i) = \frac{\mathbf{q}_{i+1} - \mathbf{q}_{i-1}}{2\Delta t}, \quad i = 0,..,N \tag{28}$$

$$\ddot{\mathbf{q}}_i = \ddot{\mathbf{q}}_i(t_i) = \frac{\mathbf{q}_{i+1} - 2\mathbf{q}_i + \mathbf{q}_{i-1}}{\Delta t^2}, \quad i = 0,..,N \tag{29}$$

where Δt is the time interval defined in Eq. (25).

If some of the state variables, such as the joint angles $\mathbf{q}$, are treated as variables in the optimization formulation, the DEs (8) are transformed to Eq. (23). The implicit problem becomes explicit. Therefore special design sensitivity analysis procedures are not needed, when the direct collocation or differential inclusion methods are employed. The finite difference approximations are used to replace the joint velocities and accelerations in terms of the displacements. Note that Eqs. (28) and (29) are the additional equality constraints between the variables.

Note that the quality of the final solution depends on the approximation made between the state variables. In this research, the central finite difference method is used to approximate the relationship between joint angles, velocities and accelerations. In order to have good results, the number of grid points usually needs to be large. However, the finite difference approximation is simple and easy to implement. This is the major advantage of the formulation. It is clear that other alternative formulations are possible. If the joint velocities $\dot{\mathbf{q}}$ or accelerations $\ddot{\mathbf{q}}$ are also treated as variables, it will give choice of expressing some constraints, e.g., the equations of motion, in terms of velocities or accelerations, to simplify their expressions. This may lead to simpler gradient evaluation and computer implementation (Wang and Arora 2004).

PIECEWISE POLYNOMIAL INTERPOLATION

In these formulations, the expansion of state and control variables are based on piecewise continuous polynomials. Linear, quadratic or cubic splines are used as the interpolating polynomials over each time segment.

Piecewise cubic 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 state equations (8) or (9) satisfied at t_i and t_{i+1}. For $t \in [t_i, t_{i+1}]$, the approximation of state variables $\mathbf{q}$ is

$$\mathbf{q}_a(t) = \sum_{k=0}^{3} \mathbf{c}_k^i \left(\frac{t - t_i}{\Delta t} \right)^k \tag{30}$$

Where

$$\begin{aligned} \mathbf{c}_0^i &= \mathbf{q}_i \\ \mathbf{c}_1^i &= \Delta t \dot{\mathbf{q}}_i \\ \mathbf{c}_2^i &= -3\mathbf{q}_i - 2\Delta t \dot{\mathbf{q}}_i + 3\mathbf{q}_{i+1} - \Delta t \dot{\mathbf{q}}_{i+1} \\ \mathbf{c}_3^i &= 2\mathbf{q}_i + \Delta t \dot{\mathbf{q}}_i - 2\mathbf{q}_{i+1} + \Delta t \dot{\mathbf{q}}_{i+1} \end{aligned} \tag{31}$$

where $\Delta t = t_{i+1} - t_i$. The approximation function (30) of the state variables must satisfy the DEs (8) or (9) at the grid point t_i $(i = 0,..,N)$. The time dependent constraints are satisfied at the grid points, and at the centers $t_{c,i} = (t_i + t_{i+1})/2$ $(i = 0,..,N-1)$. The optimization problem is to determine $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\boldsymbol{\tau}$ to minimize the cost function of Eq. (14), subject to the equality constraints in Eqs. (8) or (9) at the discrete time grid points, and the discretized inequality constraints in Eq. (24) at the time grid point and the center of each subinterval. At the grid point $t = t_i$,

$$\ddot{\mathbf{q}}_i = \ddot{\mathbf{q}}_a(t_i) = \frac{2}{\Delta t^2}\left(-3\mathbf{q}_i + 3\mathbf{q}_{i+1} - 2\Delta t \dot{\mathbf{q}}_i - \Delta t \dot{\mathbf{q}}_{i+1}\right) \tag{32}$$

A system of equations is obtained at each time grid point, by pluging $\mathbf{q}$, $\dot{\mathbf{q}}$ and Eq. (32) into the discretized equations of motion in Eq. (23) (Hargraves and Paris 1987; von Stryk and Bulirsch 1992). However, additional relationships between the joint angle and velocity variables can be obtained from Eqs. (30) and (31).

Cubic B-spline interpolation (B-spline)

If the final state or control variable history needs to be very smooth, uniform cubic B-spline interpolation (Mortenson 1985) or *NURBS* (Piegl and Tiller 1997) can be applied. B-splines have many important properties such as continuity, differentiability, and local control. These properties, especially differentiability and local control, make B-splines be competent to represent joint trajectories, which require smoothness and flexibility. There are a number of ways to define the B-spline basis functions, for simplicity of presentation the uniform B-spline is used. 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 evenly spaced for a uniform B-spline. A cubic B-spline is defined as

$$q(t) = \sum_{j=0}^{nct} N_{j,4}(t) P_j; \quad 0 \le t \le T \tag{33}$$

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 non-periodic knot vector (($m+1$) knots). For $t \in [t_i, t_{i+1}]$, using a parameter $u \in [0,1]$, such that

$$t = t_i + u\Delta t \tag{34}$$

the basis functions of a cubic B-spline are as follows

$$\begin{aligned} N_{j,4}(u) &= \tfrac{1}{6}\left(-u^3 + 3u^2 - 3u + 1\right) \\ N_{j+1,4}(u) &= \tfrac{1}{6}\left(3u^3 - 6u^2 + 4\right) \\ N_{j+2,4}(u) &= \tfrac{1}{6}\left(-3u^3 + 3u^2 + 3u + 1\right) \\ N_{j+3,4}(u) &= \tfrac{1}{6}\left(u^3\right) \end{aligned} \tag{35}$$

Since each segment of the curve is defined by four control points, for $t_i \le t_{i+1}$, Eq. (33) can be simplified as

$$q(u) = N_{i,4}(u)P_i + N_{i+1,4}(u)P_{i+1} + N_{i+2,4}(u)P_{i+2} + N_{i+3,4}(u)P_{i+3} \tag{36}$$

Since the first and second order derivatives of the displacement are needed in the dynamic optimization, the kth derivatives of a cubic B-spine curve can be easily obtained from Eq. (33), since only the basis functions are functions of time. In this formulation, the control points $\mathbf{P}$ for each DOF are chosen as the optimization variables. The equality and inequality constraints are those defined in Eqs. (26) and (27).

SERIES EXPANSION

This is to use a series of expansion in terms of orthogonal polynomials. Globally orthogonal polynomials such as Legendre and Chebshev polynomials or Lagrange polynomials can be used to discretize the state and control variables. Polynomial approximation of the state variables can be written as

$$\mathbf{q}^M(t) = \sum_{j=0}^{M} \mathbf{P}_j \phi_j(t) \tag{37}$$

where $\phi_j(t)$ are the Lagrange interpolating polynomials of order M, and they are functions of the jth order Chebyshev polynomials expressed by

$$T_j(t) = \cos\left(j \cos^{-1} t\right), \quad t \in [-1,1], \quad j = 0,..,M \tag{38}$$

The original control time interval needs to be transformed to the interval $[-1,1]$. In the Chebyshev pseudospectral method, the interpolation (grid) points are (Fahroo and Ross 2002)

$$t_k = \cos\left(\frac{\pi k}{M}\right), \quad k = 0,..,M \tag{39}$$

EVALUATION OF THE FORMULATIONS

Advantages and disadvantages of different formulations are listed in Table 2. Since the objective and constraint functions are all explicit in terms of the optimization variables in the direct collocation or differential inclusion formulation, 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 a certain time points in the direct collocation formulations. The equations of motion in Eq. (8) or (9) 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 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. These approaches are well used in trajectory optimization, chemical process engineering, and robotic motion planning. 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.

In terms of the discretization techniques, several different ways can be used: direct discretization by finite differences, piece-wise polynomials of various orders, splines, and series of expansion. A relative comparison of these techniques is listed in Table 3. All the discretization methods are applicable for first order and second order DEs. For the piece-wise polynomial interpolation or series expansion, the numbers of grid points N and variables are usually not very large; therefore, the resulting NLP is not too large. Sparsity can be utilized in the formulations, but not necessary. Some of the formulations provide good smoothness for the final solution. Therefore, they are well suitable for trajectory design, digital human motion planning, etc. However, the drawback of these formulations is that the implementation is not straightforward. They are sometimes too restrictive; therefore no solution or good solution can be obtained. Since it is a general case to have small time steps to guarantee convergence and stability for the finite difference method, the number of grid points N is usually very large, resulting in very large numbers of variables and constraints. Large-scale NLP algorithms with sparse matrix capabilities are required to solve the problems efficiently. These methods are more suitable for responses which may not necessarily be smooth.

NUMERICAL EXAMPLES

One simple numerical example is presented for clarifying the basic ideas and formulations presented in the paper. Different formulations are solved using a dense and a sparse SQP algorithm in SNOPT (Gill *et al.* 2003). Results of the examples are listed and compared. Advantages and disadvantages of the formulations are discussed. A Dell PC with 2.53 GHz processor and 1.0 GB RAM is used for running the programs and recording the relative CPU times.

A 2 DOF ARM MODEL

A two-link rigid manipulator arm is considered and solved using the formulations developed in the previous section. The reason to use this example is to present basic ideas and to validate the numerical results, since the solutions of this well-studied example are readily available. The two links are considered rigid and the joint coordinates are selected as independent generalized coordinates. The manipulator under study consists of two links whose lengths are L_1 and L_2 and moments of inertia I_1 and I_2. θ_1 and θ_2 are the relative joint angels that are controlled by the joint actuator torques τ_1 and τ_2. The manipulator is assumed to lie in the horizontal plane, and the gravity effects are neglected in writing the equations of motion. The equations of motion for the two-link manipulator are given by Dissanayake *et al.* (1991), Goh *et al.* (1993), Kota (1996) and Furukawa (2002). The parameters of the manipulator are given as $L_1 = L_2 = 0.4$ m, $m_1 = m_2 = 0.5$ kg, and $I_1 = I_2 = 0.1$ kg-m^2. The upper and lower bounds for control torques are ± 10 Nm. Three situations are considered in this example.

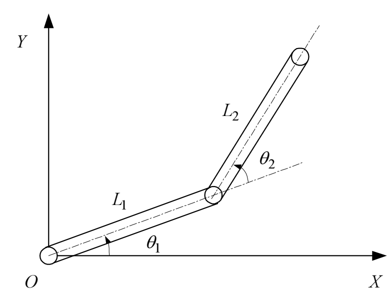

Figure 3. A two-link arm model

<u>Case 1</u>

The OCP is formulated such that the end point of the manipulator is desired to move form one location to another in minimum time. Note that the joint actuator torques τ_1 and τ_2 must be determined such that the end point moves from a position $(\theta_{10}, \theta_{20})$ in the horizontal plane at time $t = 0$ to a point $(\theta_{1T}, \theta_{2T})$ at time $t = T$. T is the final time that needs to be minimized. The OCP is subjected to the following initial and final conditions:

$$\theta_{10} = 0.0, \theta_{20} = -2.0, \dot{\theta}_{10} = 0.0, \dot{\theta}_{20} = 0.0 \quad (40)$$

$$\theta_{1T} = 1.0, \theta_{2T} = -1.0, \dot{\theta}_{1T} = 0.0, \dot{\theta}_{2T} = 0.0 \quad (41)$$

where θ_{10}, θ_{20}, θ_{1T} and θ_{2T} are the initial and final values of the joint angles that specify the orientation of the links. The terminal velocities are taken as zero. Central difference, Hermite interpolation and B-spline are used in this example. The number of variables, constraints and the final minimum time are listed in Table 4. The minimum time is similar to that reported in the literature (0.3945 s by Furukawa (2002)). Note that in table 4, N is the number of time intervals considered. $\mathbf{P}$ contains the unknown control points, and nct is the number of control points, respectively. Figures 4, 5 and 6 show the motion, joint angles and torques of the 2-link arm, respectively.

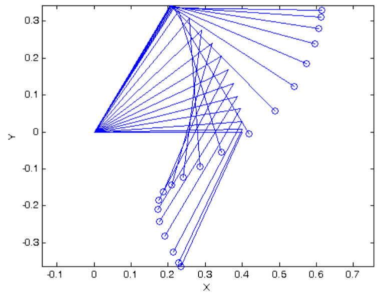

Figure 4. Motion of Case 1

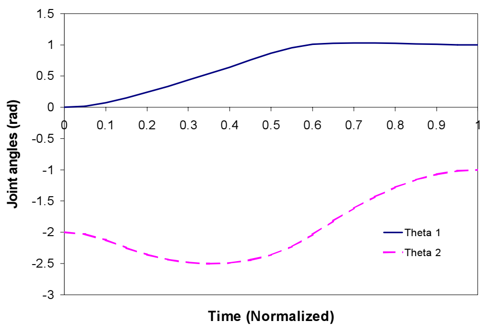

Figure 5. Joint angles of Case 1

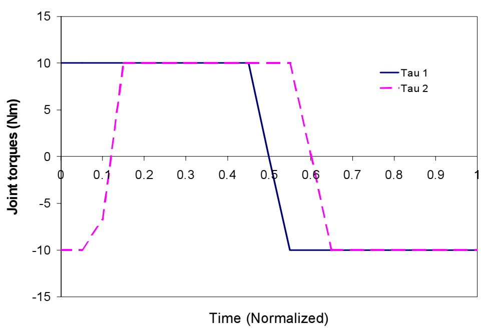

Figure 6. Joint torques of Case 1

Case 2

Additional point-wise constraints on the state variables are considered, as

$$g = |\theta_2| - 2.0 \le 0.0 \quad (42)$$

The inclusion of point-wise state constraints brings more nonlinear constraints in the state elimination formulation, and more linear constraints in the B-spline based formulations. However, in others formulations these constraints are simple bounds, as shown in Table 5. The motion, joint angles and torques of the 2-link arm are shown in figures 7, 8 and 9, respectively. The minimum time is better than that reported in the literature (0.424 s by Kota (1996)).

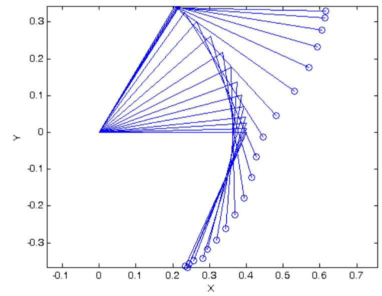

Figure 7. Motion of Case 2

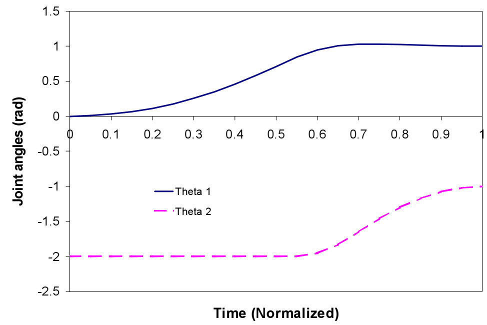

Figure 8. Joint angles of Case 2

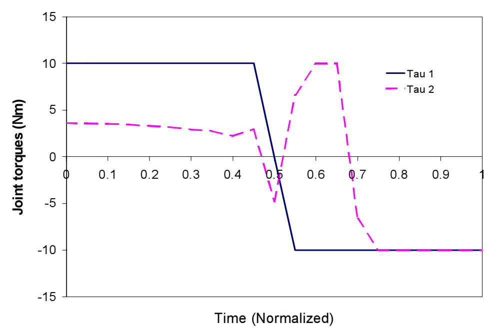

Figure 9. Joint torques of Case 2

Case 3

In this section, the two-link manipulator is considered for trajectory tracking (Kota 1996). The problem is described such that the controller must determine the joint actuator torques required to follow a specified trajectory. The objective of this problem is therefore to minimize the penalty for deviating the trajectory given by

$$J = \int_0^1 \alpha \left(y_{\text{actual}} - y_{\text{desired}}(x_{\text{end}}) \right)^2 dt \quad (43)$$

where x_{end} is the *x*-coordinate of the end point, y_{actual} and y_{desired} are the actual and desired *y*-coordinates of the end point and is a scaling parameter. A total duration of $T = 1.0$ second is considered in solving this problem. Figure 10 shows the motion.

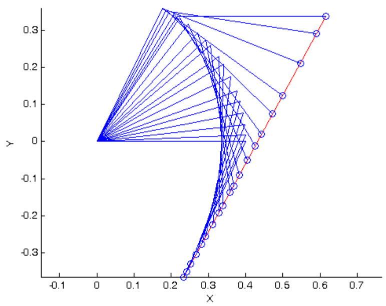

Figure 10. Motion of Case 3

OTHER EXAMPLES

Some large-scale human models are currently under investigation. The differential inclusion formulation based on B-spline curve approximations has been applied to a 30 DOF upper body model and a gait analysis model, which will be presented as two separate papers in the 2005 SAE DHM Conference.

DISCUSSION OF RESULTS

It is seen that various formulations work for the example problem and similar optimal solutions are obtained. Table 4 and 5 list the variables, sizes, numbers of iterations, and CPUs for all the formulations. It is seen that it generally takes more time for the state variable elimination formulation to find the optimal point, since repeated integration of the equations of motion is needed. Formulations based on direct collocation and differential inclusion generally require less CPU effort. Differential inclusion includes fewer variables than direct collocation. The inclusion of point-wise state constraints, such as the joint angle limits brings more constraints in the state elimination formulation, and more linear constraints in the B-spline based formulations. However, it shows no special difficulty for others formulations based on direct collocation and differential inclusion. For the formulations based on collocation and differential inclusion, it is seen that the problems are quite spase, with large numbers of zeros in the constraint Jacobian. The consideration of matrix sparsity can in general reduce the data storage by an order of magnitude. The sparse NLP code works quite well for these formulations, and turns to be the key to solve large-scale problems.

CONCLUSION

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 critically presented and compared. Different discretization techniques were discussed and their strength and weakness compared. It was shown that direct collocation and differential inclusion methods had great potential for virtual human motion prediction, because the equations of motions did not need to be integrated in the optimization process. It turns out that these methods provide new horizon for optimization-based motion prediction. All functions of the formulations became explicit in terms of the optimization variables. Derivatives of the functions with respect to the variables were computed explicitly. By introducing more variables into the formulations, the forms of the constraints and their derivatives were changed in the formulations based on direct collocation. Therefore sparse optimization algorithms for large numbers of variables and constraints were used. The formulations were illustrated by a well-studied sample problem and their solutions also compared with those available in the literature. However, these numerical methods for task-based motion prediction still need further evaluation and validation for large-scale human motion prediction. More work is needed to fully develop nonlinear optimization-based control techniques, such as direct collocation and 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 differential inclusion formulation based on B-splines is currently being implemented in the virtual human environment SANTOS^{TM}. Some further research also includes comparison with other models now being used, and the consideration of collision avoidance in the formulation.

ACKNOWLEDGMENTS

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

REFERENCES

Abdel-Malek, K.; Yu, W.; Jaber, M.; Duncan, J. 2002: Realistic posture prediction for maximum dexterity. *SAE Transactions J. Passenger Cars-Mechanical Sytems*, **110**(6), 2241-2249.

Albuquerque, J.S.; Biegler, L.T. 1997: Decomposition algorithms for on-line estimation with nonlinear DAE models. *Comp. & Chem. Eng.* **21**(3), 283-299

Alexander, R.M. 1997: A minimum energy cost hypothesis for human arm trajectories. *Biological Cybernetics*, **76**, 97-105.

Anderson, F.C.; Ziegler, J.M.; Pandy, M.G.; Whalen, R.T. 1995: Application of high-performance computing to simulation of human movement. *J. Biomechanical Eng., Transactions of the ASME,* **117**, 155-157.

Atkinson, K.E. 1988: An Introduction to Numerical Analysis. New York: John Wiley and Sons.

Arora, J.S. 1999: Optimization of structures subjected to dynamic loads. In: Leondes, C.T. (ed.) *Structural Dynamic Systems Computational Techniques and Optimization: Optimization Techniques*, pp. 1-72. The Netherlands: Gordon and Breach Science Publishers.

Arora, J.S. 2004: *Introduction to Optimum Design*, *Second Edition*. Elsevier Academic Press.

Audu, M.L.; Davy, D.T. 1988: Comparison of optimal

control algorithms for complex bioengineering studies. *Optim. Cont. Appl. & Meth.* **9**(1), 101-106.
Bathe, K.-J. 1982: *Finite Element Procedures in Engineering Analysis*. Englewood Cliffs, New Jersey: Prentice-Hall Inc.
Betts, J.T. 1990: Sparse Jacobian updates in the collocation method for optimal control problems. *J. Guidance, Cont. Dyn.* **13**(3), 409-415
Betts, J.T.; Huffman, W.P. 1992: Application of sparse nonlinear programming to trajectory optimization. *J. Guidance, Cont. Dyn.* **15**(1), 198-206.
Betts, J.T. 1998: Survey of numerical methods for trajectory optimization. *J. Guidance, Cont. Dyn.* **21**(2), 193-207.
Betts, J.T. 2000: Very low-thrust trajectory optimization using a direct SQP method. *J. Comp. Appl. Math.* **120**(1), 27-40.
Betts, J.T.; Biehn, N.; Campbell, S.L.; Huffman, W.P. 2000: Compensating for order variation in mesh refinement for direct transcription methods. *J. Comp. Appl. Math.* **125**(1-2), 147-158
Betts, J.T.; Biehn, N.; Campbell, S.L.; Huffman, W.P. 2002: Compensating for order variation in mesh refinement for direct transcription methods II: computational experience. *J. Comp. Appl. Math.* **143**(2), 237-261
Biegler, L.T. 1988: On the simultaneous solution and optimization of large scale engineering systems. *Comp. Chem. Eng.* **12**, 357-369.
Biegler, L.T.; Ghattas, O.; Heinkenschloss, M.; Bloemen Waanders, B.v. (Eds.) 2003: *Large-Scale PDE-Constrained Optimization, Lecture Notes in Computational Science and Engineering*, Vol. 30. Berlin: Springer-Verlag.
Bottasso, C.; Croce, A. 2004: Optimal control of multibody systems using an energy preserving direct transcript method. *Multibody Sys. Dyn.* **12**, 17-45.
Bryson, A.E.; Ho, Y.C. 1975: *Applied Optimal Control*. New York: Hemishper Publishing
Cervantes, A.; Biegler, L.T. 1998: Large-scale DAE optimization using a simultaneous NLP formulation. *AIChE J.* **44**(5), 1038-1050.
Cervantes, A.M.; Wächter, A.; Tütüncü, R.H.; Biegler, L.T. 2000: A reduced space interior point strategy for optimization of differential algebraic systems. *Comp. & Chem. Eng.* **24**(1), 39-51.
Chaffin, D.B. 1997: Development of computerized human static strength simulation model for job design. *Human Factors & Ergonomics in Manufacturing*, **7**, 305-322.
Chen, Y.-C. 1991: Solving robot trajectory planning problems with uniform cubic B-splines. *Optim. Cont. Appl. & Meth.* **12**, 247-262.
Chow, C.K.; Jacobson, D.H. 1971: Studies of human locomotion via optimal programming. *Math. Biosci.* **10***,* 293-306.
Conway, B.A.; Larson, K.M. 1998: Collocation versus differential inclusion in direct optimization. *J. Guidance, Cont. Dyn.* **21**(5), 780-785.
Coverstone-Carroll, V.; Williams, S.N. 1994: Optimal low thrust trajectories using differential inclusion concepts. *J. Astron. Sci.* **42**(4), 379-393.
Denavit, J. and Hartenberg, R.S., 1955: A Kinematic notation for lower-pair mechanisms based on matrices. *J. Applied Mechanics*, **77**, 215-221.
Dissanayake, M.W.M.G.; Goh, C.J.; Phan-Thien, N. 1991: Time-optimal trajectories for robot manipulators. *Robotica*, **9**, 131-138.
Enright, P.J.; Conway, B.A. 1992: Discrete approximation to optimal trajectories using direct transcription and nonlinear programming. *J. Guidance, Cont. Dyn.* **15**(4), 994-1002.
Fahroo, F.; Ross, I.M. 2002: Direct trajectory optimization by a Chebyshev pseudospectral method. *J. Guidance, Cont. Dyn.* **25**(1), 160-166.
Faraway, J.J.; Zhang, X.D.; Chaffin, D.B. 1999: Rectifying postures reconstructed from joint angles to meet constraints. *J. Biomechanics*, **32**, 733-736.
Furukawa, T. 2002: Time-subminimal trajectory planning for discrete non-linear systems. *Eng. Optim.* **34**, 219-243.
Gill, P.E.; Murray, W.; Saunders, M.A. 2002: SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM J. Optim.* **12**, 979-1006.
Goh, C.J.; Edwards, N.J.; Zomaya, A.Y. 1993: Feedback control of minimum-time optimal control problems using neural networks. *Optim. Cont. Appl. & Meth.* **14**, 1-16.
Haftka, R.T. 1985: Simultaneous analysis and design. *AIAA J.* **23**(7), 1099-1103.
Hargraves, C.R.; Paris, S.W. 1987: Direct trajectory optimization using nonlinear programming and collocation. *J. Guidance, Cont. Dyn.* **10**(4), 338-342.
Hase, K.; Yamazaki, N. 1997: Development of three-dimensional whole-body musculoskeletal model for various motion analyses. *JSME Int. J.*, Series C, **40**(1), 25-32.
Herman, A.L.; Conway, B.A. 1995: Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules. *J. Guidance, Cont. Dyn.* **19**(3), 592-599.
Hsieh, C.C.; Arora, J.S. 1984: Design sensitivity analysis and optimization of dynamic response. *Appl. Math. & Optim*. **43**(2), 195-219.
Hu, G.S.; Ong, C.J.; Teo, C.L. 2002: An enhanced transcribing scheme for the numerical solution of a class of optimal control problems. *Eng. Optim.* **34**(2), 155-173.
Hull, D.G. 1997: Conversion of optimal control problems into parameter optimization problems. *J. Guidance, Cont. Dyn.* **20**(1), 57-60.
Hull, D.G. 2003: *Optimal Control Theory for Applications*. New York: Springer-Verlag
Jung, E.S.; Choe, J. 1996: Human reach posture prediction based on psychophysical discomfort", *Int. J. of Industrial Ergonomics*, **18**(2-3), 173-179.
Kaplan, M.L.; Heegaard J.H. 2001: Predictive algorithms for neuromuscular control of human locomotion. *J. Biomechanics*, **34**(8), 1077-1083.
Kaplan, M.L.; Heegaard, J.H. 2002: Second-order optimal control algorithm for complex systems. *Int. J. Num. Meth. Eng.* **53**(9), 2043-2060.
Khang, G.; Zajac, F.E. 1989a: Paraplegic standing

controlled by functional neuromuscular stimulation - I: Computer model and control-system design. *IEEE Transactions on Biomedical Eng.* **36**(9), 873-884.

Khang, G.; Zajac, F.E. 1989b: Paraplegic standing controlled by functional neuromuscular stimulation - II: Computer simulation studies. *IEEE Transactions on Biomedical Eng.* **36**(9), 885-894.

Kim, J.H. 2004: *Human performance measures*, Technical Report No. VSR-04.02. Virtual Soldier Research program/CCAD, The University of Iowa, Iowa City, IA 52242.

Kota, N.N. 1996: *Computational Methods for Nonlinear Optimal Control*. Ph.D. Dissertation, Department of Mechanical Engineering, University of Iowa, Iowa City 52242.

Kraft, D. 1985: On converting optimal control problems into nonlinear programming codes. In: Schittkowski, K. (ed.) *Computational Mathematical Programming*, NATO ASI Series 15, pp. 261-280. Berlin: Springer.

Kumar, R.R.; Seywald, H. 1996: Should controls be eliminated while solving optimal control problems via direct methods? *J. Guidance, Cont. Dyn*. **19**(2), 418-423

Leineweber, D.B.; Bauer, I.; Bock, H.G.; Schloder, J.P. 2003: An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part 1: Theoretical aspects. *Comp. & Chem. Eng.* **27**(2), 157-166

Lo, J.; Huang, G.; Metaxas, D. 2002: Human motion planning based on recursive dynamics and optimal control techniques. *Multibody Sys. Dyn.* **8**(4), 433-458.

Luo, Z.-Q.; Pang, J.-S.; Ralph, D. 1996: *Mathematical Programs with Equilibrium Constraints*. Cambridge: Cambridge University Press.

Marler, R.T. 2004: *Development of real-time multi-objective optimization-based posture prediction*, Technical Report No. VSR-04.02. Virtual Soldier Research program/CCAD, The University of Iowa, Iowa City, IA 52242.

Marler, R.T., and Arora, J.S. 2004: Survey of multi-objective optimization methods for engineering. *Struct. Multidisc. Optim.* **26**(6), 369-395.

Mi, Z. 2004: *Task-based Prediction of Upper Body Motion*. PhD Dissertation, Department of Mechanical Engineering, University of Iowa, Iowa City, IA 52242.

Mortenson, M.E. 1985: *Geometric Modeling*, New York, NY: John Wiley & Sons, Inc.

Neptune, R.R; Hull, M.L. 1998: Evaluation of performance criteria for simulation of submaximal steady-state cycling using a forward dynamic model. *J. Biomechanical Eng.* **120**, 334-340.

Pandy, M.G.; Zajac, F.E.; Sim, E.; Levine, W.S. 1990: Optimal control model for maximum-height human jumping. *J. Biomechanics*, **23**(12), 1185-1198.

Pandy, M.G.; Anderson, F.C.; Hull, D.G. 1992: A Parameter optimization approach for the optimal control of large-scale musculoskeletal systems. *J. Biomechanical Eng., Transactions of the ASME,* **114**(4), 450-460.

Pandy, M.G.; Garner, B.A.; Anderson, F.C. 1995: Optimal control of non-ballistic muscular movements: A constraint-based performance criterion for rising from a chair. *J. Biomechanical Eng., Transactions of the ASME*, **117**(1), 15-26.

Paul, R.P. 1981: *Robot Manipulators: Mathematics, Programming, and Control.* Cambridge, Massachusetts: MIT Press.

Piegl, L.; Tiller, W. 1997: *The NURBS Book, 2nd Edition*. New York, NY: Springer-Verlag.

Pontryagin, L.S.; Boltyanskii, V.; Gamkrelidze, R.; Mishchenko, E. 1962: *The mathematical Theory of Optimal Process*. New York: Interscience Publishers Inc.

Ross, I.M.; Fahroo, F. 2004: Pseudospectral knotting methods for solving optimal control problems. *J. Guidance, Cont. Dyn.* **27**(3), 397-405.

Seywald, H. 1994: Trajectory optimization based on differential inclusion. *J. Guidance, Cont. Dyn.* **17**(3), 480-487.

Schulz, V. 2004: Simultaneous solution approaches for large optimization problems. *J. Comp. Appl. Math.* **164**(1), 629-641.

Tanartkit, P.; Biegler, L.T. 1996: Reformulating ill-conditioned differential - algebraic equation optimization problems. *Indus. & Eng. Chem. Res.* **35**(6), 1853-1865

Tashman, S.; Zajac, F.E.; Perkash, I. 1995: Modeling and simulation of paraplegic ambulation in a reciprocating gait orthosis. *J. Biomechanical Eng., Transactions of the ASME*, **117**(3), 300-308.

Tseng, C.H.; Arora, J.S. 1989: Optimum design of systems for dynamics and controls using sequential quadratic programming. *AIAA J.* **27**, 1793-1800.

Vlassenbroeck, J. 1988: A Chebyshev polynomial method for optimal control with state constraints. *Automatica*, **24**(4), 499–506.

von Stryk, O.; Bulirsch, R. 1992: Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, **37**, 357-373.

Wang, C.-Y.E.; Timoszyk, W.K.; Bobrow, J.E. 2001: Payload maximization for open chained manipulators: Finding weightlifting motions for a Puma 762 robot. *IEEE Trans. Robot. & Autom.* **17**(2), 218-224.

Wang, Q.; Arora, J.S. 2004: Alternate formulations for transient dynamic response optimization. *AIAA J.* (under review).

Williams, P. 2004: Jacobi pseudospectral method for solving optimal control problems. *J. Guidance, Cont. Dyn.* **27**(2), 293-297.

Yang, J.; Marler, R.T.; Kim, H.; Arora, J.; Abdel-Malek, K. 2004: Multi-objective optimization for upper body posture prediction. *10th AIAA/ISSMO Multi-disciplinary Analysis and Optimization Conf.*, August, Albany, NY.

CONTACT

Corresponding author: J.S. Arora. Email: jasbir-arora@.uiowa.edu; telephone: (319) 335-5658; fax: (319) 335-5660.

Table 1 Different formulations for motion prediction

Design variable : $\boldsymbol{\tau}$ Minimize : Single or multiple human performance measures Subject to : $\mathbf{q}^L \le \mathbf{q}(\boldsymbol{\tau}) = \mathbf{f}^{-1}(\boldsymbol{\tau}) \le \mathbf{q}^U$ (forward dynamics) end - effector Bounds on variables : $\boldsymbol{\tau}^L \le \boldsymbol{\tau} \le \boldsymbol{\tau}^U$	Design variable : $\mathbf{q}, \boldsymbol{\tau}$ Minimize : Single or multiple human performance measures Subject to : $\boldsymbol{\tau} - \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \mathbf{0}$ (simultaneous) end - effector Bounds on variables : $\mathbf{q}^L \le \mathbf{q} \le \mathbf{q}^U$ $\boldsymbol{\tau}^L \le \boldsymbol{\tau} \le \boldsymbol{\tau}^U$	Design variable : $\mathbf{q}$ Minimize : Single or multiple human performance measures Subject to : $\boldsymbol{\tau}^L \le \boldsymbol{\tau}(\mathbf{q}) = \mathbf{f}(\mathbf{q}) \le \boldsymbol{\tau}^U$ (inverse dynamics) end - effector Bounds on variables : $\mathbf{q}^L \le \mathbf{q} \le \mathbf{q}^U$
$\boldsymbol{\tau}^{\text{initial}} \rightarrow$ Optimizer $\rightarrow \boldsymbol{\tau}^{\text{optimal}}$ $\downarrow$ (f.d.) $\mathbf{q}(\boldsymbol{\tau}^{\text{optimal}}) = \mathbf{f}^{-1}(\boldsymbol{\tau}^{\text{optimal}})$	$\boldsymbol{\tau}^{\text{initial}}, \mathbf{q}^{\text{initial}} \rightarrow$ Optimizer $\rightarrow \boldsymbol{\tau}^{\text{optimal}}, \mathbf{q}^{\text{optimal}}$	$\mathbf{q}^{\text{initial}} \rightarrow$ Optimizer $\rightarrow \mathbf{q}^{\text{optimal}}$ $\downarrow$ (i.d.) $\boldsymbol{\tau}(\mathbf{q}^{\text{optimal}}) = \mathbf{f}(\mathbf{q}^{\text{optimal}})$
State variable elimination	Direct collocation	Differential inclusion

Table 2 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 3 Advantages and disadvantages of different discretization techniques

Discretization Techniques	Variables	Advantages	Disadvantages
Finite Difference	State or control variables at discretized points	• 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	State or control variables at discretized points or coefficients of polynomial expansion	• Good smoothness. • Smaller NLP problems. • State or control variable constraints become linear in most cases.	• The required curve profile may be too restrictive. • Implementation sometimes is not straightforward.
Series Expansion	Coefficients of series expansion	• Good smoothness. • Smaller NLP problems • State or control variable constraints become linear in most cases.	• The required curve profile may be too restrictive. • Implementation sometimes is not straightforward.

Table 4 Design example (Case 1)

Formulations		Variables	No. of variables	No. of Constraints	No. of Non-zeros in Constraint Jacobian (w. Sparsity)	No. of Non-zeros in Constraint Jacobian (w/o Sparsity)	No. of Iteration	T (s)	CPU (s)
State Variable Elimination		$T, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2$	$2N+3$	4	$8N+12$	$8N+12$	21	0.393	6.72
Direct Collocation	**CD**	$T, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2$	$4N+9$	$2N+6$	$16N+24$	$8N^2+42N+36$	10	0.392	0.05
	Hermite	$T, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dot{\boldsymbol{\theta}}_1, \dot{\boldsymbol{\theta}}_2, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2$	$6N+11$	$4N+2$	$32N+20$	$24N^2+56N+22$	37	0.393	0.57
	B-spline	$T, \mathbf{P}_{\theta_1}, \mathbf{P}_{\theta_2}, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2$	$2nct+2N$ $+3$	$2N+10$	$20N+44$	$4N^2+4Nnct+26N$ $20nct+30$	22	0.394	0.19
Differential Inclusion	**CD**	$T, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2$	$2N+7$	$2N+6$	$14N+22$	$4N^2+16N+42$	15	0.392	0.09
	Hermite	$T, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dot{\boldsymbol{\theta}}_1, \dot{\boldsymbol{\theta}}_2$	$4N+9$	$4N+2$	$30N+18$	$16N^2+44N+18$	316	0.393	3.81
	B-spline	$T, \mathbf{P}_{\theta_1}, \mathbf{P}_{\theta_2}$	$2nct+1$	$2N+10$	$18N+42$	$4Nnct+2N$ $+20nct+10$	16	0.394	0.16

Table 5 Design example (Case 2)

Formulations		Variables	No. of variables	No. of Constraints	No. of Non-zeros in Constraint Jacobian (w. Sparsity)	No. of Non-zeros in Constraint Jacobian (w/o Sparsity)	No. of Iteration	T (s)	CPU (s)
State Variable Elimination		$T, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2$	$2N+3$	$N+4$	$2N^2+11N+12$	$2N^2+11N+12$	78	0.405	22.56
Direct Collocation	**CD**	$T, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2$	$4N+9$	$2N+6$	$16N+24$	$8N^2+42N+36$	15	0.404	0.08
	Hermite	$T, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dot{\boldsymbol{\theta}}_1, \dot{\boldsymbol{\theta}}_2, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2$	$6N+11$	$4N+2$	$32N+20$	$24N^2+56N+22$	50	0.405	0.78
	B-spline	$T, \mathbf{P}_{\theta_1}, \mathbf{P}_{\theta_2}, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2$	$2nct+2N$ $+3$	$3N+11$	$24N+48$	$6N^2+6Nnct+31N$ $22nct+33$	17	0.406	0.16
Differential Inclusion	**CD**	$T, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2$	$2N+7$	$2N+6$	$14N+22$	$4N^2+16N+42$	26	0.404	0.14
	Hermite	$T, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dot{\boldsymbol{\theta}}_1, \dot{\boldsymbol{\theta}}_2$	$4N+9$	$4N+2$	$30N+18$	$16N^2+44N+18$	150	0.405	2.22
	B-spline	$T, \mathbf{P}_{\theta_1}, \mathbf{P}_{\theta_2}$	$2nct+1$	$3N+11$	$22N+46$	$6Nnct+3N$ $+22nct+11$	18	0.406	0.23