Multibody Syst Dyn (2008) 19: 427–451
DOI 10.1007/s11044-007-9100-4

A novel formulation for determining joint constraint loads during optimal dynamic motion of redundant manipulators in DH representation

Joo H. Kim · Jingzhou Yang · Karim Abdel-Malek

Received: 6 May 2007 / Accepted: 15 November 2007 / Published online: 10 January 2008

Abstract The kinematic representations of general open-loop chains in many robotic applications are based on the Denavit–Hartenberg (DH) notation. However, when the DH representation is used for kinematic modeling, the relative joint constraints cannot be described explicitly using the common formulation methods. In this paper, we propose a new formulation of solving a system of differential-algebraic equations (DAEs) where the method of Lagrange multipliers is incorporated into the optimization problem for optimal motion planning of redundant manipulators. In particular, a set of fictitious joints is modeled to solve for the joint constraint forces and moments, as well as the optimal dynamic motion and the required actuator torques of redundant manipulators described in DH representation. The proposed method is formulated within the framework of our earlier study on the generation of load-effective optimal dynamic motions of redundant manipulators that guarantee successful execution of given tasks in which the Lagrangian dynamics for general external loads are incorporated. Some example tasks of a simple planar manipulator and a high-degree-of-freedom digital human model are illustrated, and the results show accurate calculation of joint constraint loads without altering the original planned motion. The proposed optimization formulation satisfies the equivalent DAEs.

Keywords Denavit–Hartenberg representation · Fictitious joints · Optimization · Motion planning · Redundant manipulator · Lagrange multipliers · Joint constraints · Differential-algebraic equations

1 Introduction

Redundancy in robotics can be defined in various ways [8]. In this paper, a redundant manipulator is defined as a manipulator with larger degrees of freedom (DOF) than required to accomplish a given task. Thus, redundant manipulators can possess an infinite number of

J.H. Kim (✉) · J. Yang · K. Abdel-Malek
US Army Virtual Soldier Research Program, Center for Computer-Aided Design,
The University of Iowa, Iowa City, IA 52242, USA
e-mail: joo-kim@uiowa.edu

configurations at one time for performing an assigned task. The redundancy of robots provides higher flexibility, dexterity, manipulability, controllability, and singularity-avoidance. Good examples are seen in the recent developments of humanoids, bio-inspired robots, and space robots.

The recent advancement in optimal motion planning methodologies for redundant manipulators is largely due to the implementation of dynamics principles, which results in optimal dynamic motions. Just like any other multibody system in motion, redundant manipulators are always subject to constraint loads (forces and moments)—or sometimes called reaction loads—between multiple connected links. Since the information on the constraint loads is critical for robotic design and dynamics analysis, methodologies that calculate accurate constraint loads for manipulators in general motion with externally applied loads are essential. Similar statements can be made for human dynamic motion, which can be predicted using the optimization-based motion planning method [19]. The determination of constraint loads during human motion is useful for injury prediction and stress analysis of joints and joint replacements.

A kinematic constraint between two bodies imposes conditions on the relative motion between the pair of bodies. Consider a system described by n generalized coordinates: $q_1, q_2, \dots, q_n$. Suppose there are m independent equality constraints and l inequality constraints as algebraic equations represented in terms of the generalized coordinates and time.

$$\phi_j(q_1, q_2, \dots, q_n, t) = 0 \quad (j = 1, \dots, m), \tag{1}$$

$$f_k(q_1, q_2, \dots, q_n, t) \le 0 \quad (k = 1, \dots, l). \tag{2}$$

Constraints of this form are known as holonomic kinematic constraints [13]. The constraints that cannot be expressed in the form of (1) or (2), but must be expressed in terms of differentials of the coordinates and/or time are known as nonholonomic constraints. Our presentation is limited to the relative joint constraints, which are holonomic, that occur at the joints of multibody systems.

Under external load conditions (including gravity), constraint loads due to the relative constraints are exerted to the rigid-body pairs. For example, a joint that connects the rigid links of an open-loop system usually generates constraint loads (Fig. 1). The directions of the constraint loads are those in which the relative motion of the links is not allowed. Similarly, human motions always encounter constraints from internal sources; thus, for a human body under external loads, constraint loads always exist at each joint.

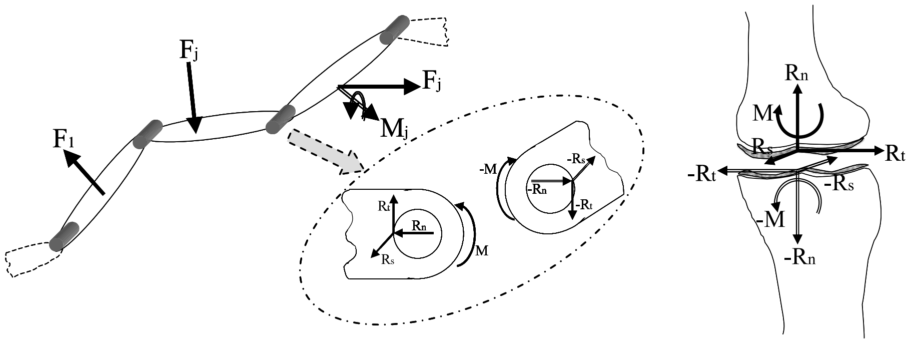


Fig. 1 Joint constraint forces and moments of an open-loop system and a human body

Determining the joint constraint loads has long been one of the main issues in the multibody dynamics area. Many successful theories and methods have been developed for general-purpose simulation and analysis of multibody mechanisms, and several commercial software packages are available (e.g., ADAMS, DADS). The basic starting point for associating constraint loads with the equations of motion is the method of Lagrange multipliers, which has been introduced in many dynamics texts (e.g., [13]). The constraint loads are usually obtained as solutions of the differential-algebraic equations (DAEs), which are formulated from a set of constrained equations of motion for a system and the constraint equations. However, the analytical complexity of nonlinear algebraic equations of kinematics and nonlinear differential equations of dynamics makes it impossible to obtain closed-form solutions in most applications. The literature in this field, therefore, contains a large number of specialized numerical techniques and analytical methods for solutions of DAEs of multibody systems [15, 24]. For example, absolute (Cartesian) coordinate formulation describes the configuration and constraints of each rigid body represented by a body-fixed reference frame with respect to a global (inertial) frame. On the other hand, the relative (joint) coordinate formulation describes the relative configurations of pairs of rigid bodies that are connected by joints, and thus the number of constraint equations is reduced. To improve computational efficiency and numerical performance, methods of coordinate partitioning have also been introduced [15, 24], in which the system coordinates are partitioned as independent and dependent variables. Other numerical analysis and methods for solving DAEs, as well as the fundamentals, are given by Brenan et al. [7] and Hairer et al. [14].

Several up-to-date numerical methods for solving DAEs were reviewed by Cuadrado et al. [10]. Examples were tested using different methods for each, and guidelines were presented as to which modeling methods are most adequate for different types of multibody systems. The formulations are very general, so they can be applied to different mechanical systems with various cases. Numerical integration approaches for DAEs using the Runge–Kutta method and its modification have also been widely studied in the literature [18, 23].

An extensive overview of the various state-of-the-art multibody dynamics research to date with historical background is given in an article by Schiehlen [25], in which the basics and the applications to broad areas are also discussed. Several improvements to the traditional methods have been made recently. For example, Blajer [5] derived pseudoinverse matrices to the constraint Jacobian by employing reduced-dimension formulations with which the joint constraint loads are obtained directly in resolved forms without matrix inversion, thus improving efficiency. Hemami and Wyman [16] introduced a slightly different approach to dealing with the constraint loads as well as the actuator forces by employing the state space form. They developed stable feedback inverse systems to estimate the forces and moments of rigid-body systems subject to holonomic and nonholonomic constraints. However, despite the numerous efforts in the literature so far, the complexity of the numerical evaluations of DAEs for highly nonlinear systems has yet to be improved.

In addition to the rigid-body motion, the deformations of constrained mechanical systems are studied as flexible multibody dynamics, and a comprehensive review of the past and recent work is given by Shabana [27]. In our presentation, we consider only the rigid-body dynamics.

As an alternative method of solving for the joint constraint loads in robotic systems, Baysec and Jones [3] introduced “the method of fictitious degrees of freedom.” They modeled a three-link chain with a revolute and a prismatic joint at each link, which actually behaves like various types of 3-DOF manipulators by assigning numerous combinations of revolute and prismatic real joints. The remaining joint at each link is the fictitious joint, which is controlled to have zero velocity and acceleration during the motion. Then the resulting actuator forces and moments of the fictitious joints are the constraint forces and moments.

The model used for demonstration is simple such that it is relatively straightforward to derive closed-form Lagrangian equations of motion and explicit expressions for the constraint loads. In their approach, the control errors due to noise signals, overshoot, damping, etc. can provide inaccurate values of constraint loads. In addition, the number of fictitious joints is not general enough to model all possible components of the constraint loads.

Recently, many of the theoretical and applicative studies on multibody dynamics (including this article) have been conducted in conjunction with human motion dynamics. McLean et al. [22] used a variable step fourth-order Runge–Kutta method to solve the forward dynamics problem for the joint reaction forces and moments at the human knee during dynamic motion. The muscle stimulation patterns as well as the initial conditions are given as inputs, where the muscle forces are calculated from the muscle stimulation via a muscle activation dynamics model. The resulting lower-body motions were reliable compared with the experimental measurement. In their method, the motion and the other inputs such as the muscle forces are not fully predictive, as those values are obtained through optimization of the measured data. As another rigorous formulation, Blajer and Czaplicki [6] proposed a compact and systematic way of determining joint constraint loads in human multibody system. They applied the previously introduced method [5] that does not involve matrix inversion to the human dynamics model, which is thus suited for both symbolic and computational implementations. The augmented joint coordinate method is introduced as a combination of the open-constraint coordinates for prohibited relative joint motions and the traditional joint coordinates. The method can be used to determine only some joint reactions. Some other groups used the inverse dynamics approach to obtain the joint constraint loads for prescribed motions. For example, Hirashima et al. [17] used the Newton–Euler method to derive the equations of motion for human model. Some simple motions of shoulder, elbow, and wrist joints are given as inputs. A unique application of the joint constraint force calculation is introduced by Biscarini and Cerulli [4]. They combined the dynamical and hydrodynamical model to calculate the knee joint constraint forces during underwater knee extension exercises. The analytical form of the constraint forces at the knee joint is derived as functions of hydrodynamic parameters (such as drag) as well as the joint profiles. Then the problem of inverse dynamics is solved for a given range of motion.

The Denavit–Hartenberg (DH) notation [11], which will be briefly described in the following section, provides an effective way of formulating the kinematics of general open-loop chains and is, therefore, one of the most widely used kinematic representation methods in robotics [9, 26]. Although much research has been conducted on DAEs, the common formulations for explicit descriptions of relative joint constraints cannot be applied if the system is represented by the DH notation and homogeneous transformation matrices. This is because the DH notation assumes that there is only one relative DOF (either revolute or prismatic) between two connected links, and the remaining DOFs are naturally constrained. In this article, we propose a method that allows calculating the joint constraint loads during an optimal dynamic motion of redundant manipulators described by the DH representation. To the authors' knowledge, there is no proposed formulation or numerical method in the current literature that deals with this kind of problem. Various methods of multibody dynamics in the literature are developed based on the main idea of incorporating the joint constraint loads into the DAEs by relaxing the kinematically constrained DOFs at each joint and constraining those that are not allowed to move by Lagrange multipliers. In our approach, we will apply this main idea to the multibody systems represented in DH notation. The contribution and advantages of our method over the traditional ones can be summarized as follows:

(1) The proposed approach allows formulating the constrained dynamics of multibody systems that are modeled based on DH representation, and the constraint loads at specified

joints or links can be determined. The process of extending and modifying the DH parameters at the points of interest and shifting the indices of the succeeding joints can be easily automated.

(2) Usually, motion planning of redundant manipulators is formulated as an optimization problem. Our proposed method is applicable within the framework of this usual optimal motion planning by providing additional control variables and constraints (zero-displacement) to the original optimization problem. The joint constraint loads are calculated concurrently during the stage of motion planning, which traditionally has been done as a post-process (numerical integration) according to the resulting motion and actuator torque profiles. It is not necessary to formulate a separate integration-based DAE solver or to call interactive multibody dynamics software; thus, it will eventually reduce computational cost and time. Also, the proposed method allows using the joint constraint loads to form certain cost functions or constraints for the optimization problem. For example, the optimal motion can be obtained such that the magnitude of the constraint loads at a specific joint is minimized or is kept below certain maximum strength. Furthermore, it will be shown later that the implementation of joint constraint loads and the addition of fictitious joints to the formulation do not affect the original results of the planned motion.

In the following sections, the DH kinematic modeling and the dynamic equations of motion with general external loads for open-loop kinematic chains are briefly overviewed. Then the structure and components of the optimal dynamic motion planning problem are presented. Next, the modeling of fictitious joints, the corresponding DAEs, and the extended form of optimization problem are introduced. Finally, some examples of determining joint constraint loads during optimal motion generation using the proposed method will be illustrated and discussed.

2 Kinematic modeling of open-loop chains by DH representation

The analysis of general robotic systems is always concerned with the configurations, velocities, and accelerations of objects (e.g., robotic links, tools, and environments) in a three-dimensional space. Given an open-loop kinematic chain (Fig. 2), the transform

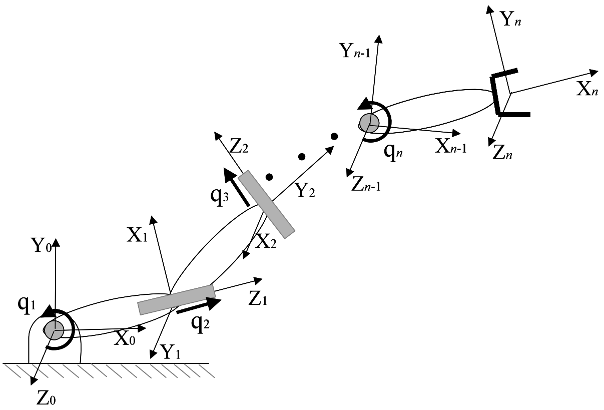


Fig. 2 An n-DOF open-loop kinematic chain

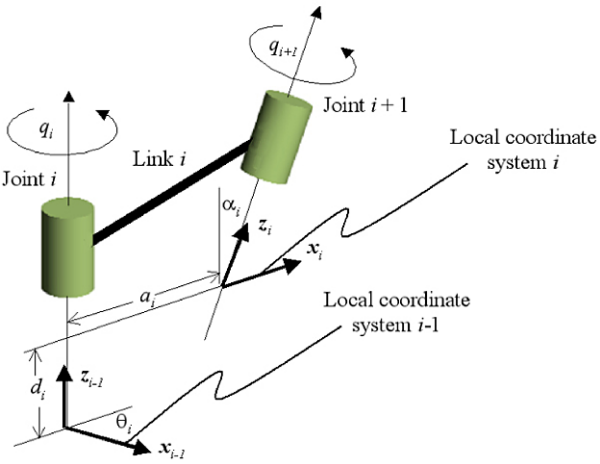


Fig. 3 DH convention and parameters

from frame $\{i\}$ to frame $\{i-1\}$ can be represented by the homogeneous transformation matrix ${}^{i-1}\mathbf{T}_i$

$$ {}^{i-1}\mathbf{T}_i(\theta_i, d_i, a_i, \alpha_i) = \begin{pmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (3) $$

If the joint is revolute, θ_i is called the joint variable, and the other three quantities, d_i, α_i, and a_i, are called link parameters. If the joint is prismatic, the joint variable is d_i, while the other three are the link parameters [9]. The convention is based on DH notation, which describes the configuration of a kinematic chain (Fig. 3).

The homogeneous transform includes the translation and rotation of one coordinate frame relative to another, where each frame is attached, respectively, to a rigid body in space. The homogeneous transformation matrix ${}^0\mathbf{T}_n$ that relates frame $\{n\}$ to frame $\{0\}$ can be obtained by multiplying all of the intermediate transforms

$$ {}^0\mathbf{T}_n(q_1, \ldots, q_n) = {}^0\mathbf{T}_1(q_1)\,{}^1\mathbf{T}_2(q_2)\ldots{}^{n-1}\mathbf{T}_n(q_n), \quad (4) $$

where q_i is the joint variable of ${}^{i-1}\mathbf{T}_i$, and the set of joint variables $\mathbf{q} = [q_1, \ldots, q_n]^{\mathrm{T}} \in \mathbf{R}^n$ is called the $n \times 1$ joint vector. Therefore, ${}^0\mathbf{T}_n$ is a function of all n joint variables. These joint variables uniquely determine the configuration of a manipulator system with n DOFs and are called the generalized coordinates. Then the position vector of a point of interest attached to the frame $\{n\}$ of the end-effector can be written with respect to the global frame $\{0\}$ using the joint variables.

For the purpose of mathematical modeling, each actual kinematic joint of a system is replaced with a set of one or more single-DOF (revolute or prismatic) joints. The DH parameters and the joint variable limits are assigned in such a way that the mobility associated with the original joint is preserved.

3 Lagrange's equations of motion

To generate the motions of a manipulator in which the externally applied forces and moments are taken into account, it is essential to formulate a comprehensive expression of the equations of motion that govern the dynamics of open-loop kinematic chains. The implementation of the equations of motion is also necessary for control of the manipulators, although the control problems are not discussed here. In this presentation, we use Lagrangian dynamics, which provides a systematic method of formulating the equations of motion that are described in terms of the independent generalized coordinates. The use of Lagrangian dynamics also allows relatively easy implementation of any kinematic constraints and the corresponding constraint loads in a systematic manner.

For the motion of a mechanical system with finite DOFs, the Hamilton's principle in a given time interval $[t_0, t_1]$ is formulated as the following variational equation [21]:

$$\delta \int_{t_0}^{t_1} (T + W)\,\mathrm{d}t = 0, \tag{5}$$

where T is the total kinetic energy of the system, W is the virtual work of the noninertial forces, and δ denotes the first variation. Both T and W are functions of $\mathbf{q}$, $\dot{\mathbf{q}}$, and t. For a holonomic system, it can be shown [20] that the solution $\mathbf{q}(t)$ of this problem satisfies the following general form of Lagrange's equations of motion (in vector-matrix form)

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} + \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial W_{\mathrm{nc}}}{\partial \dot{\mathbf{q}}} - \frac{\partial W_{\mathrm{nc}}}{\partial \mathbf{q}} = \mathbf{0}, \tag{6}$$

where $L = T - V$ is the Lagrangian function, V is the total potential energy of the system, and W_{nc} is the virtual work done by nonconservative forces.

Let us consider the case in which a general form of external loads $[\mathbf{F}_k^{\mathrm{T}}\ \mathbf{M}_k^{\mathrm{T}}]^{\mathrm{T}}$ is applied to the point at ${}^k\mathbf{r}_k$ location of link k, where $[\mathbf{F}_k^{\mathrm{T}}\ \mathbf{M}_k^{\mathrm{T}}]^{\mathrm{T}}$ is a 6×1 vector comprised of a 3×1 force vector $\mathbf{F}_k$ and a 3×1 moment vector $\mathbf{M}_k$, and ${}^k\mathbf{r}_k$ is a 4×1 position vector expressed in terms of $\{k\}$ local coordinate frame attached to link k. Note that the constraint forces and moments due to external constraints from the environment can also be expressed in this manner. Equation (6) can be expanded using the kinetic energy, the potential energy, and the extended form of non-conservative work (for details, see [19]). Assuming that the velocity-dependent force does not exist, i.e., $\mathrm{d}(\partial W_{\mathrm{nc}}/\partial \dot{\mathbf{q}})/\mathrm{d}t = 0$, the final vector-matrix form of the equations of motion for a general open-loop kinematic chain with general external loads is given as a coupled, nonlinear, second-order ordinary differential equation

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) + \sum_i \mathbf{J}_i^{\mathrm{T}} m_i \mathbf{g} + \sum_k \mathbf{J}_k^{\mathrm{T}} \begin{bmatrix} -\mathbf{F}_k \\ -\mathbf{M}_k \end{bmatrix} + \mathbf{T}(\mathbf{q}, \dot{\mathbf{q}}), \tag{7}$$

where $\boldsymbol{\tau} = [\tau_1, \tau_2, \ldots, \tau_n]^{\mathrm{T}}$ is the actuator torque vector, $\mathbf{M}(\mathbf{q})$ is the mass-inertia symmetric matrix, $\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis and centrifugal force vector, $\sum \mathbf{J}_i^{\mathrm{T}} m_i \mathbf{g}$ is the joint torque vector due to gravity, $\mathbf{J}_i$ is the Jacobian matrix of the position vector for the center of mass of ith link, and $\mathbf{J}_k$ is the augmented Jacobian matrix of the position vector ${}^k\mathbf{r}_k$ with respect to $\{k\}$ local coordinate frame. The vector $\mathbf{T}(\mathbf{q}, \dot{\mathbf{q}})$ is the torque vector due to the joint stiffness and the dissipative forces such as viscous damping and Coulomb friction. The augmented Jacobian matrix $\mathbf{J}_k(\mathbf{q})$ is derived from the linear relationship between the tangent spaces of the joint variables and the Cartesian coordinates

$$\mathbf{J}_k(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_{k,1}(\mathbf{q}) & \ldots & \mathbf{J}_{k,i}(\mathbf{q}) & \ldots & \mathbf{J}_{k,k}(\mathbf{q}) \end{bmatrix}_{6\times k}, \tag{8}$$

where the ith column vectors for revolute and prismatic joints are, respectively,

$$\mathbf{J}_{k,i}^{\text{revolute}}(\mathbf{q}) = \begin{bmatrix} \frac{\partial^0\mathbf{T}_k(\mathbf{q})}{\partial q_i}\,{}^k\mathbf{r}_k \\ {}^0\mathbf{z}_{i-1}(\mathbf{q}) \end{bmatrix}_{6\times 1} \quad \text{and} \quad \mathbf{J}_{k,i}^{\text{prismatic}}(\mathbf{q}) = \begin{bmatrix} \frac{\partial^0\mathbf{T}_k(\mathbf{q})}{\partial q_i}\,{}^k\mathbf{r}_k \\ \mathbf{0}_{3\times 1} \end{bmatrix}_{6\times 1}.$$

Here, ${}^0\mathbf{z}_{i-1}$ $(i = 1, \dots, k)$ is the unit z-axis vector of $\{i-1\}$ local frame expressed in terms of the global coordinate frame. Note that in the above equation, only the first three elements of the 4×1 vector $(\partial^0\mathbf{T}_k(\mathbf{q})/\partial q_i)^k\mathbf{r}_k$ are used to assemble the first three rows of $\mathbf{J}_{k,i}(\mathbf{q})$.

4 Optimal dynamic motion planning for redundant manipulators

The problem of optimal dynamic motion planning for redundant manipulators is defined as follows (Fig. 4): The inputs to the algorithm are the link parameters of the manipulator, dynamic parameters (such as mass, centers of mass, moments and products of inertia, joint stiffness, and damping coefficients), joint variable limits, actuator torque limits (possibly as functions of joint velocity), points of application and components of external loads, and task-based constraints (such as the time desired to perform the task, end-effector path and orientations). Then it is desired to generate the joint profiles that guarantee the execution of the task, where the external loads can have broad ranges of magnitudes. To resolve the redundancy, the problem is formulated as an optimization problem, where the outputs are the joint variable profiles, the required actuator torques, and the energy rates as functions of time. The proposed optimal motion planning problem is stated as:

Find: Joint control points ($\mathbf{P}_{(nc\times n)}$)
To minimize: Energy consumption (E)
Subject to constraints:
  Joint limits ($\mathbf{q}^L \le \mathbf{q} \le \mathbf{q}^U$)
  Actuator torque limits ($\boldsymbol{\tau}^L \le \boldsymbol{\tau} \le \boldsymbol{\tau}^U$)
  Path constraints ($\|\mathbf{x}(\mathbf{q}(t)) - \mathbf{path}(t)\| \le \varepsilon$).

It should be emphasized that the general external loads term as well as the inertia and gravity, must be included in the calculation of actuator torques that are used for the constraints (torque limits) and/or the cost function. The load-effective motions are defined as the feasible motions in which the general external loads are taken into account for the calculation of required actuator torques to guarantee the execution of the planned motion [20]. The energy consumption or the norm of torque vector used as the cost function will result in the

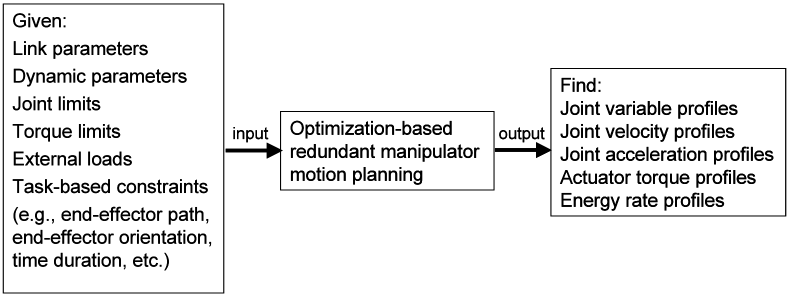


Fig. 4 Problem of optimal motion planning for redundant manipulators

most efficient motion. If the optimization is solved while the external loads are not considered for the dynamics calculation, then the actual required actuator torques may exceed the maximum torque limits that the actuators can provide; thus, the planned motion may not be executable (i.e., uncontrollable) in reality, especially when large external loads are desired.

Due to the redundancy of the human body, the approach for human motion prediction is basically the same as that described for the redundant manipulators. This is based on the assumption that humans naturally generate effective motion to accomplish a given task in such a way as to minimize certain cost function(s). Note that the problem for manipulators is "motion planning," while for humans it is "motion prediction." For convenience, we will use the term "motion generation" for both manipulators and humans. The details of each component of the optimization problem are described in the following sections.

For the numerical optimization algorithm, we use the sequential quadratic programming (SQP) method. The SQP uses quasi-Newton approximations to the Hessian of the augmented Lagrangian and obtains search directions from a sequence of quadratic programming subproblems. SQP methods have proved reliable and highly effective for solving constrained optimization problems with smooth nonlinear cost function and constraints. The details on the SQP method can be found in optimization texts (e.g., [2]).

4.1 Joint variable profiles using B-spline curves

Since the joint variables as functions of time are nonuniform curves, we use the B-spline curves [1], which have many beneficial properties such as continuity, differentiability, endpoint interpolations, local control, and convex hull. We use the recursive formula to represent the B-spline, such that its control points will be calculated as a result of the iterative numerical optimization algorithm. Let *nk* be the number of knots and $\mathbf{U} = \{u_0, \dots, u_{nk-1}\}$ be a knot vector with a nondecreasing sequence of knots. The ith B-spline basis function of p-degree (order $p+1$), denoted by $N_{i,p}(u)$, is defined as

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{if } u_i \le u < u_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \\ N_{i,p}(u) &= \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u). \end{aligned} \tag{9}$$

To enforce endpoint interpolations and continuity, we choose a $(nk \times 1)$ nonperiodic knot vector that has multiplicities at the start and the end as follows:

$$\mathbf{U} = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{nk-p-2}, \underbrace{b, \dots, b}_{p+1}\}.$$

Then the pth-degree B-spline curve for each joint can be written as

$$q_j(u) = \sum_{i=0}^{nc-1} N_{i,p}(u) P_{i,j} \quad (a \le u \le b;\ j = 1, \dots, n), \tag{10}$$

where n is the total DOF of the system, *nc* is the number of control points, and $\{P_{i,j}\}$ represents the (*i, j*) components of the control points matrix $\mathbf{P}_{(nc \times n)}$. Here, the degree p, the number of the control points *nc*, and the number of knots *nk* are related by $nk = nc + p + 1$.

The degree and the multiplicity of the knots of the B-spline curve determine the continuity and differentiability. A smooth joint motion requires continuity in acceleration, which will in turn require the joint B-spline curve to be at least of degree-3, i.e., $p = 3$. Let the initial time $a = 0$ and the final time $b = t_f$, and a total of 11 distinct knots are used: $0, 0.1t_f, 0.2t_f, \ldots,$ and t_f. Therefore, $nk = 17$ and $nc = 13$, i.e., each joint B-spline curve has 13 control points.

4.2 Energy consumption

Energy has a unifying property into which the dynamic as well as the kinematic characteristics of manipulator motion are incorporated. We use an approximate form of manipulator energy consumption as a cost function for our optimization problem. The actual formula representing the energy consumption for a manipulator varies depending on the specific design of the system, as well as the types of actuators. Therefore, unless the details of the specific machine information and physical characteristics are provided, it is not possible to obtain the exact formula for the energy consumption. For this reason, simplified forms of the general energy consumption are widely used in literature. Usually, the energy consumption is modeled to be proportional to the actuator torques. We use the squared norm of the actuator torque vector function as an approximate form of the energy consumption from time t_1 to t_2

$$E = \left\| \boldsymbol{\tau}(t) \right\|^2 = \int_{t_1}^{t_2} \sum_{i=1}^{n} \bigl(\tau_i(t)\bigr)^2 \, \mathrm{d}t, \tag{11}$$

where the actuator torque vector $\boldsymbol{\tau}(t) = [\tau_1(t), \ldots, \tau_n(t)]^{\mathrm{T}}$ is obtained from the equations of motion.

The use of energy consumption as a cost function implies several important points. First of all, minimum energy consumption indicates minimum fuel usage. Secondly, for smooth movement of each joint, the magnitude of the second derivatives of the joint curves needs to be minimized to avoid an abrupt change in the joint velocity. The joint accelerations (second derivatives) term that constitutes the actuator torques in (7) provides a natural way to ensure the smooth movement of each joint by reducing unnecessary fluctuations in the joint profiles.

For human motion prediction, the metabolic energy consumption that was derived previously [19] will be used as a cost function for the optimization problem

$$E_{\mathrm{Metabolic}} \approx \int_{t_1}^{t_2} \sum_{i=1}^{n} \bigl|\tau_i(t)\dot{q}_i(t)\bigr| \, \mathrm{d}t + \int_{t_1}^{t_2} \sum_{i=1}^{n} h_m^i \bigl|\tau_i(t)\bigr| \, \mathrm{d}t + \int_{t_1}^{t_2} \dot{B} \, \mathrm{d}t, \tag{12}$$

where h_m^i $(i = 1, \ldots, n)$ are the coefficients of the generalized maintenance heat and $\dot{B}$ is the basal metabolic rate. It was shown that h_m^i is inversely proportional to the maximum torque limit of joint i. Therefore, for small joint velocities, the human motion of minimum energy (thus, minimum weighted torques) implies that humans tend to use the stronger joints to accomplish a given task rather than the weaker ones. This means that the actuator torques are distributed so that the larger torques are exerted at the stronger joints and vice versa, which can be observed in real-world human tasks.

4.3 Constraints

The following is a list of basic constraints that are typically given from the manipulator design and the task requirements. Depending on the task definition and the environment, various other constraints can be imposed in addition.

(1) Joint variable limits Each joint variable has bilateral constraints imposed in the form of

$$q_i^{\mathrm{L}} \le q_i \le q_i^{\mathrm{U}} \quad (i = 1, \ldots, n), \tag{13}$$

where q_i^{L} and q_i^{U} are the lower and upper limits for each joint variable, respectively. These joint limits are usually given from the design of the manipulator.

(2) Actuator torque limits The torque limit is usually a function of the joint velocity, which is represented as a torque-speed curve of each actuator. The torque-speed curves depend on the class and capacity of the actuators and are usually supplied by the manufacturer

$$\tau_i^{\mathrm{L}}(\dot{q}_i(t)) \le \tau_i \le \tau_i^{\mathrm{U}}(\dot{q}_i(t)) \quad (i = 1, \ldots, n), \tag{14}$$

where τ_i^{L} and τ_i^{U} are the lower and upper limits, respectively, for each actuator torque. Generally, τ_i^{L} is negative and τ_i^{U} is positive.

(3) Position and orientation constraints In general, the configuration of a single rigid body in space is uniquely determined in terms of three independent position coordinates and three independent orientation angles. Thus, the configuration of a link of a manipulator system can be described uniquely by assigning its position and orientation. Depending on the task requirements, some of these six coordinates can be constrained, while the rest are left as free DOFs. For position constraints, the Cartesian coordinates of the point as a function of the joint variables are constrained. For orientation constraints, the direction of the unit vectors of the link local frame is constrained in terms of the global frame.

(4) Path constraints Every manipulator motion generates an end-effector path along the time in Cartesian space. This end-effector path may be either constrained by task requirements or naturally unconstrained. Usually, the path is given as a task requirement. For example, the end-effector paths of drawing a straight line or welding on a surface are predetermined from the task requirements. Suppose the end-effector path for the task is assigned as a parametric curve in Cartesian space such as

$$\mathbf{path}(t) = \left[x_{\mathrm{path}}(t),\ y_{\mathrm{path}}(t),\ z_{\mathrm{path}}(t)\right]^{\mathrm{T}}. \tag{15}$$

To ensure that the end-effector point characterized by $\mathbf{x} = [x, y, z]^{\mathrm{T}}$ as a function of joint variables stays on the path during the motion, the distance from the end-effector point to the desired path in the Cartesian space is enforced as a constraint

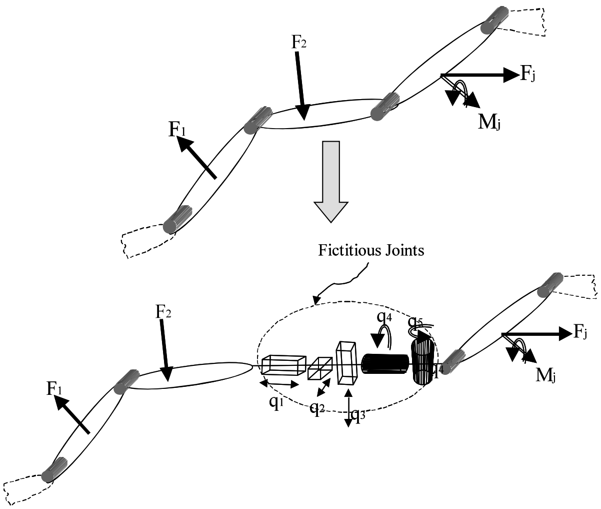
$$\left\| \mathbf{x}\big(\mathbf{q}(t)\big) - \mathbf{path}(t) \right\| \le \varepsilon, \tag{16}$$

where $0 \le t \le t_f$ and ε is a small positive number as a specified tolerance (e.g., 0.001).

5 Modeling of fictitious joints

In a three-dimensional space, the relative configuration between two unconstrained rigid bodies has six DOFs. In a general robotic motion, certain constraint loads exist at joints because some or all of the six relative DOFs between the two connected links are not allowed, i.e., they are constrained. This feature is equivalent to a system with two links connected by six joints that represent the six relative DOFs, with some of these constrained. Based

Fig. 5 Modeling of fictitious joints



on this concept, additional joints are modeled besides the real joints to formulate the joint constraint loads. When the DH representation method is used, a single joint represents one DOF, which is either revolute or prismatic. Therefore, for a single joint, the remaining five DOFs are constrained between the two links. We add these five joints to the real joint. If the real joint is revolute, then the other five are composed of two revolute and three prismatic joints (Fig. 5). If the real joint is prismatic, then the other five are composed of two prismatic and three revolute joints. Since these are not the real joints, they are called fictitious joints.

Consider an open-loop system of n generalized coordinates. Let f be the number of fictitious joints at the point of interest. For convenience and without loss of generality, denote the additional fictitious joint variables as $q_{n+1}, q_{n+2}, \dots, q_{n+f}$. Because of the addition of fictitious joints, the total number of DOFs is now increased to $n + f$. It is thus necessary to enforce additional constraints, the number of which is same as the number of fictitious joints. In reality, the fictitious joints cannot generate any displacements. Thus, the additional constraints are the zero-displacement constraints at the fictitious joints. Mathematically, these are the equality constraints expressed as follows:

$$^{\text{fict}}\boldsymbol{\Phi} = \begin{bmatrix} q_{n+1} \\ \vdots \\ q_{n+f} \end{bmatrix}_{f\times 1} = \mathbf{0}. \tag{17}$$

Then the actuator forces and torques corresponding to the fictitious joints are the joint constraint loads. In the context of constrained dynamics, the joint constraint loads can be obtained from coordinate transformations of the Lagrange multipliers $\lambda_1, \dots, \lambda_f$ associated with the constraint equation (17). In this manner, the method of Lagrange multipliers can be formulated for the systems represented by DH notation. Since the fictitious joints are not real, the link lengths, masses, and moments/products of inertia are all zero for the associated fictitious links. Note that, although these zero parameters may lead to sparsity in the matrices of the equations of motion (7), the proposed method does not encounter the usual numerical difficulties of ill-conditioning. This is because this method does not involve any forward dynamics approach, and the numerical integration or matrix inversion of the equations of motion is not required.

5.1 Formulation of DAEs

A system of DAEs can be formulated with the fictitious joints. The Jacobian matrix of the constraints (17) is calculated as follows

$$ {}^{\mathrm{fict}}\boldsymbol{\Phi}_{\mathbf{q}} = \frac{\partial}{\partial [q_1\ \dots\ q_{n+f}]^{\mathrm{T}}} \begin{bmatrix} q_{n+1} \\ \vdots \\ q_{n+f} \end{bmatrix}_{f\times 1} = \begin{bmatrix} \mathbf{0}_{f\times n} & \mathbf{I}_{f\times f} \end{bmatrix}_{f\times(n+f)}. \tag{18} $$

Here, we use the popular notation ${}^{\mathrm{fict}}\boldsymbol{\Phi}_{\mathbf{q}}$ for the constraint Jacobian matrix, while $\mathbf{J}$ has been used to represent the configuration Jacobian matrices in the equations of motion. For each generalized coordinate q_i of the system, there exists a corresponding generalized constraint force C_i. The generalized constraint force vector $\mathbf{C}$ can be obtained from the constraint Jacobian matrix and the Lagrange multipliers

$$ \mathbf{C} = \begin{bmatrix} C_1 \\ \vdots \\ C_{n+f} \end{bmatrix}_{(n+f)\times 1} = -{}^{\mathrm{fict}}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} = -\begin{bmatrix} \mathbf{0}_{n\times f} \\ \mathbf{I}_{f\times f} \end{bmatrix}_{(n+f)\times f} \begin{bmatrix} \lambda_{n+1} \\ \vdots \\ \lambda_{n+f} \end{bmatrix}_{f\times 1} = -\begin{bmatrix} \mathbf{0}_{n\times 1} \\ \lambda_{n+1} \\ \vdots \\ \lambda_{n+f} \end{bmatrix}_{(n+f)\times 1}. \tag{19} $$

The constrained equations of motion is then

$$ \boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) + \sum_i \mathbf{J}_i^{\mathrm{T}} m_i \mathbf{g} + \sum_k \mathbf{J}_k^{\mathrm{T}} \begin{bmatrix} -\mathbf{F}_k \\ -\mathbf{M}_k \end{bmatrix} + \mathbf{T}(\mathbf{q},\dot{\mathbf{q}}) + {}^{\mathrm{fict}}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} \tag{20} $$

subject to

$$ {}^{\mathrm{fict}}\boldsymbol{\Phi} = [q_{n+1}\ \dots\ q_{n+f}]_{f\times 1}^{\mathrm{T}} = \mathbf{0}. \tag{21} $$

The Lagrange multiplier form of the equations of motion (20) and the constraint equations (21) must be satisfied simultaneously at all times, which constitutes a set of mixed DAE of index-3 [7]. Let $\mathbf{Q}(\mathbf{q},\dot{\mathbf{q}})$ be the vector that contains the external and nonconservative forces as well as the velocity-dependent inertia forces as follows

$$ \mathbf{Q}(\mathbf{q},\dot{\mathbf{q}}) = \boldsymbol{\tau} - \mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) - \sum_i \mathbf{J}_i^{\mathrm{T}} m_i \mathbf{g} - \sum_k \mathbf{J}_k^{\mathrm{T}} \begin{bmatrix} -\mathbf{F}_k \\ -\mathbf{M}_k \end{bmatrix} - \mathbf{T}(\mathbf{q},\dot{\mathbf{q}}). \tag{22} $$

Then the constrained equations of motion (20) can be written as

$$ \mathbf{M}\ddot{\mathbf{q}} + {}^{\mathrm{fict}}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{Q}(\mathbf{q},\dot{\mathbf{q}}). \tag{23} $$

Equation (23), combined with the double time-derivatives of the constraint functions (21), forms the following standard matrix equation called DAE of index-1

$$ \begin{bmatrix} \mathbf{M} & {}^{\mathrm{fict}}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ {}^{\mathrm{fict}}\boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q}(\mathbf{q},\dot{\mathbf{q}}) \\ -{}^{\mathrm{fict}}\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} - {}^{\mathrm{fict}}\dot{\boldsymbol{\Phi}}_{t} \end{Bmatrix}, \tag{24} $$

where the upper dot indicates the total derivative with respect to time, and the subscript indicates partial derivatives. For a nonredundant manipulator with known actuator torques, the total number of equations is the same as the total number of unknown variables, i.e., q's and λ's. As mentioned earlier, solving these equations for nonredundant systems with $\mathbf{q}$ and $\boldsymbol{\lambda}$ as unknowns usually requires numerical integration methods [7, 15].

The above equations of motion can be simplified by noting that there is no actuator torque or force at the fictitious joints (i.e., no energy input is provided). Thus, by moving the generalized constraint load vector term in (20) to the left-hand side, we obtain

$$\begin{aligned}&\begin{bmatrix}\tau_1 & \ldots & \tau_n, -\lambda_{n+1} & \ldots & -\lambda_{n+f}\end{bmatrix}^{\mathrm{T}}_{(n+f)\times 1}\\ &\quad= \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) + \sum_i \mathbf{J}_i^{\mathrm{T}} m_i \mathbf{g} + \sum_k \mathbf{J}_k^{\mathrm{T}} \begin{bmatrix} -\mathbf{F}_k \\ -\mathbf{M}_k \end{bmatrix} + \mathbf{T}(\mathbf{q},\dot{\mathbf{q}}).\end{aligned} \tag{25}$$

If the negatives of Lagrange multipliers are regarded as the actuator torques corresponding to the fictitious joints, (25) is equivalent to the equations of motion for nonconstrained dynamics. Therefore, in practice, the fictitious joints are treated exactly like the real ones, and the routine calculation of actuator torques using optimization will solve for the constraint loads as well.

5.2 Optimization problem for motion generation and joint constraint loads determination

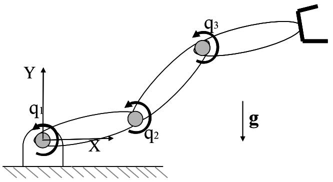
Since the joint constraint loads may vary with time, the λ's will also, in general, be functions of time. In our dynamics formulation, the λ's are evaluated at each time step. The control variables, cost function, and constraints for the optimization are the same as those described previously, except for some following differences due to the addition of the fictitious joints. The fictitious joint variables are treated like real joint variables, and the corresponding control points $\hat{\mathbf{P}}_{(nc\times f)}$ serve as additional variables. However, it is important to note that the constraint loads from the fictitious joints should not be included in the energy calculation. In other words, only the actuator torques from the real joints are used to calculate the energy consumption. Another notable difference is the addition of constraints due to the fictitious joints. The additional constraints imposed on the fictitious joints can be treated like real joint limits with both the lower and upper limits zero. Also note that unlike the real joints, the fictitious joints do not have any maximum force limit constraints. The final optimization problem for the joint constraint loads and motion planning of redundant manipulators can be stated as follows:

Find: Joint control points ($\mathbf{P}_{(nc\times n)}$) and fictitious joint control points ($\hat{\mathbf{P}}_{(nc\times f)}$)
To minimize: Energy consumption (E)
Subject to constraints:
Joint limits ($\mathbf{q}^L \le \mathbf{q} \le \mathbf{q}^U$)
Actuator torque limits ($\boldsymbol{\tau}^L \le \boldsymbol{\tau} \le \boldsymbol{\tau}^U$)
Path constraints ($\|\mathbf{x}(\mathbf{q}(t)) - \mathbf{path}(t)\| \le \varepsilon$)
Fictitious joint limits (${}^{\text{fict}}\boldsymbol{\Phi} = \mathbf{0}$).

At each time interval, the joint variables, required actuator torques, energy rates, and joint constraint loads are calculated.

6 Example results and discussion

The proposed method will be demonstrated using a 3-DOF planar manipulator (motion planning) and a 21-DOF human model (motion prediction). For each case, the optimal motion is



$$-\pi \leq q_1 \leq \pi; \; -8500 \leq \tau_1 \leq 8500$$
$$-\pi \leq q_2 \leq \pi; \; -4300 \leq \tau_2 \leq 4300$$
$$-\pi \leq q_3 \leq \pi; \; -1500 \leq \tau_3 \leq 1500$$

Fig. 6 A 3-DOF planar manipulator

Table 1 The DH parameters for the 3-DOF planar manipulator

Joint	θ	d	α	a
1	q_1	0	0	1
2	q_2	0	0	1
3	q_3	0	0	1

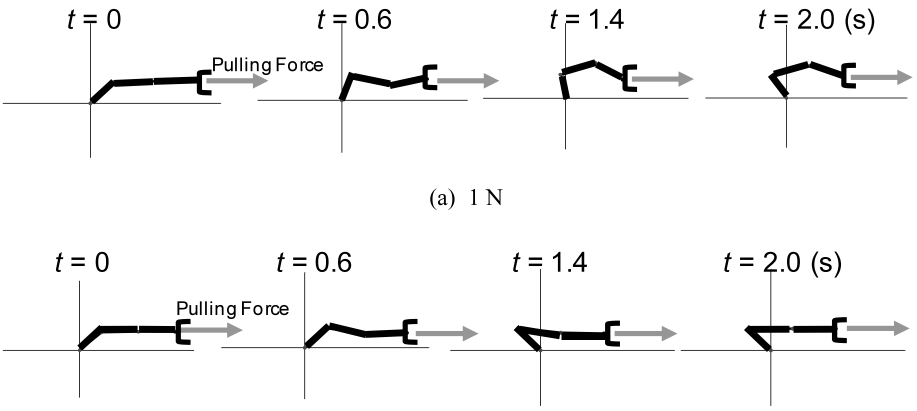
generated first without calculating the joint constraint loads. Next, the same optimal motion problem is solved with the proposed method associated with the fictitious joints, and the results will be analyzed and compared. Although simple tasks are illustrated as examples in this presentation, our method can be easily applied to more complicated tasks.

Redundant manipulator Consider a 3R planar manipulator (Fig. 6) where each link is modeled as a thin rod with 1 m of length and 10 kg of mass. The DH parameters are listed in Table 1. For the equations of motion, we neglect the torques due to the stiffness and dissipative properties at the joints. The joint variable limits (radians) and the actuator torque limits (Nm) are given as shown below.

The task is defined in the X–Y plane as follows. Given (2.6, 0.866, 0) (m) and (1.5, 0.866, 0) (m) as the initial and the final global coordinates of the end-effector, respectively, the manipulator is required to pull (or drag) in the $-X$ direction along a straight line with external forces at the end-effector. The time duration is given as 2 seconds. However, depending on the task requirements or user input, different time durations can be used. This task represents simple manipulations such as pulling an object or opening a door. To compare the resulting motions from two extreme cases, we use 1 N and 10000 N as pulling forces. The generated motions and joint profiles are shown in Figs. 7, 8, from which we can see that each joint moves smoothly toward the final position. The computing times of the optimization process for the 1-N- and 10000-N-pulling are 16.72 and 15.39 seconds, respectively.

The required actuator torques (Fig. 9) for the generated motions satisfy the torque limits, and thus guarantee the successful execution of the task. Each motion profile is a load-effective motion for the given pulling force under specified actuator capacity limits. The details are reported in our previous work [20], and some important differences from the comparison of the two results are briefly explained below.

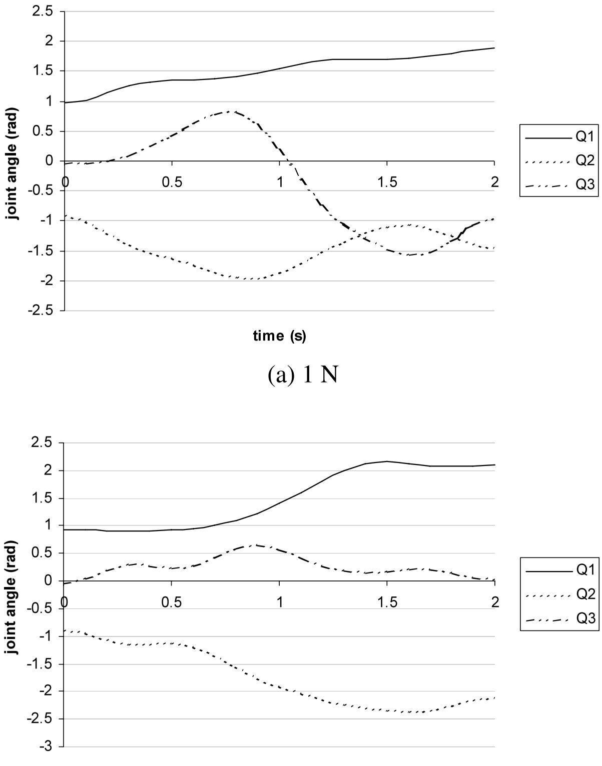
For the 1-N pulling, the positive torque values of joint 2 are mostly used to sustain the gravity. It is shown that the values of the link weights dominate the motion and the effect of the small pulling force is negligible. For the 10000-N pulling, the successful execution of the task can be explained by analyzing the manipulator configurations and the force equilibrium of the system free-body diagram at each time step. It is observed that the manipulator tries to maintain the alignment of links 2 and 3 with the line of application of the large pulling force (Fig. 7(b)); this is not notable for the small pulling force. Since the magnitude and



(a) 1 N

(b) 10000 N

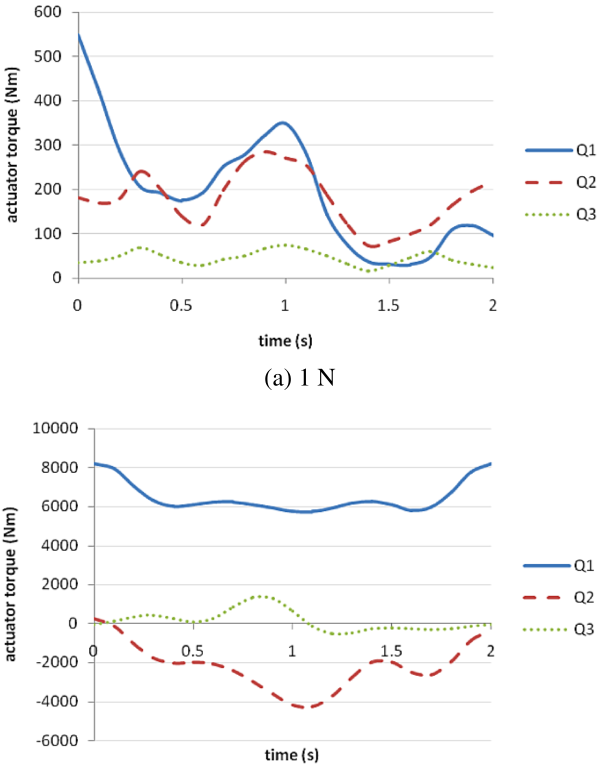
Fig. 7 Generated pulling motions of a 3-DOF planar manipulator



(a) 1 N

(b) 10000 N

Fig. 8 Generated joint profiles for pulling motion



(a) 1 N

(b) 10000 N

Fig. 9 Required actuator torque profiles for pulling motion

direction of the pulling force is constant, the actuator torque values to sustain the pulling force at joint 1 in Fig. 9(b) are almost constant (except for the initial and final stages) due to the constant moment-arm length. The actuator torques at joints 2 and 3 can be roughly calculated as the product of the constant pulling force and the offset distance perpendicular from the line of the pulling force (of course, the inertia forces and gravity should be added to obtain exact torque values). Thus, in the case of the large-force pulling task in which the effects of the link weights are relatively small, the actuator torques are mostly used to sustain the large pulling force. By positioning joints 2 and 3 as close to the line of force as possible, the actuator torque values for those joints are minimized. Note that a similar trend is observed in human arm motion when pulling with large force.

Next, let us generate the optimal motion for the same task, while calculating the joint constraint forces. In addition to the usual outputs, it is required to determine the joint constraint forces between links 2 and 3. Since the model is two-dimensional, we need two fictitious prismatic joints between links 2 and 3 to achieve the full three DOFs in the plane. Let us renumber these fictitious joints as joints 3 and 4 and the original joint 3 as joint 5 (Fig. 10). Then the DH parameters are extended to include the fictitious joints between links 2 and 3 (Table 2).

The calculated results for joint variables and required actuator torques are exactly same as our previous results (Figs. 7–9). At this point, it is important to note that the addition of the fictitious joints for joint constraint loads calculation does not affect the original opti-

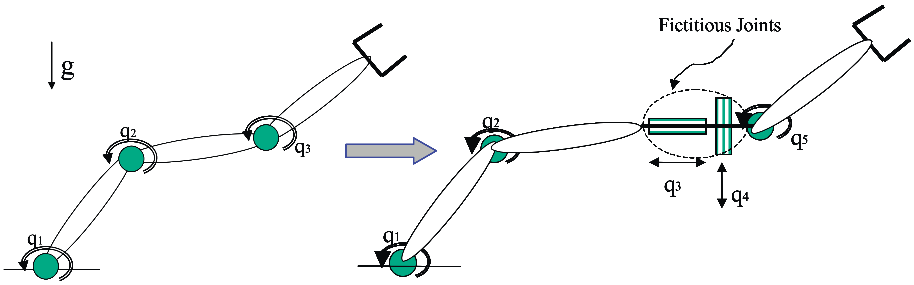


Fig. 10 The fictitious joints of a 3-DOF planar manipulator

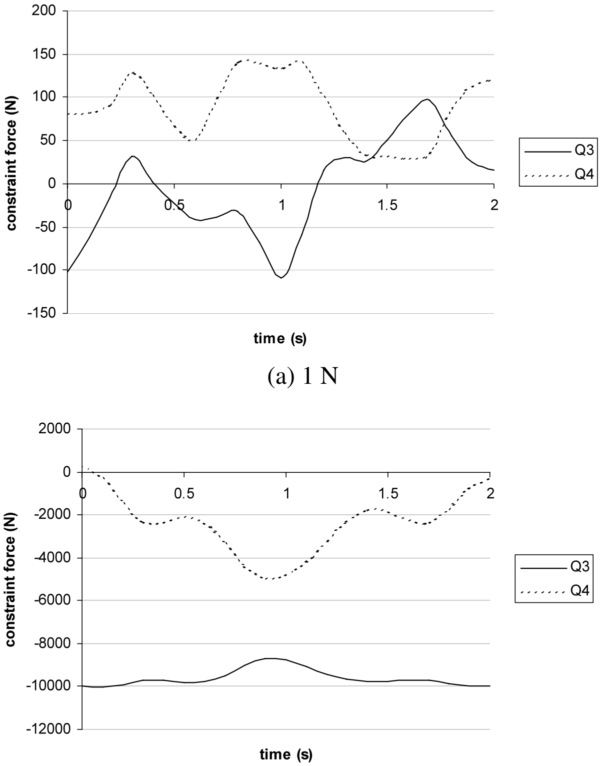
Table 2 Extended DH parameters for the 3-DOF planar manipulator

Joint	θ	d	α	a
1	q_1	0	0	1
2	$\pi/2+q_2$	0	$\pi/2$	0
3	$\pi/2$	$1+q_3$	$\pi/2$	0
4	$\pi/2$	q_4	$\pi/2$	0
5	q_5	0	0	1

mal motion results. With the extended optimization formulation, the joint constraint forces are the calculated actuator forces at joints 3 and 4 (Fig. 11). The computing times of the optimization with fictitious joints for the 1-N- and 10000-N-pulling are 149.16 and 136.84 seconds, respectively. Considering that the closed form of the Lagrangian equations of motion (7) has order of $O(N^4)$, the difference between the computing times with and without fictitious joints is plausible. However, depending on the formulation types of the equations of motion, the time difference can be further reduced (e.g., $O(N)$ for the recursive Newton–Euler method).

As a quick intuitive validation, the above results for joint constraint forces can be easily verified by writing force equilibrium for the free-body diagram of links 2 and 3. The constraint forces at joints 3 and 4 should mainly sustain the applied pulling force (1 N and 10000 N) and the weight of link 3 (and inertia forces, of course). For example, since link 3 is kept almost horizontal during the 10000-N pulling motion, it is easy to check that our result for fictitious joint 3 is approximately -10000 N throughout the motion.

For more rigorous verification of our method and results, the equations of motion (20) with given actuator torques, external loads (including gravity), and constraint loads should be integrated to obtain the joint variable profiles (forward dynamics). The initial conditions should be read from the joint variable profiles. Note that now the differential equations of motion should be solved instead of the DAEs, since the Lagrange multipliers are known, and thus the algebraic constraint equations are dropped. On the other hand, as another way to verify the solution of the equations of motion, the inverse dynamics with given joint variable profiles can be solved for the actuator torques and the constraint loads. This idea is based on the existence and uniqueness theorem for the solution of forward dynamics with fixed initial conditions. For multibody systems with no friction or contacts, there exists a unique set of forward dynamics solutions for given initial conditions. The details of the theorem and proof can be found in texts of nonlinear dynamics and differential equations (e.g., [28]). For systems with frictional contact, the existence and uniqueness criteria of the forward



(a) 1 N

(b) 10000 N

Fig. 11 Joint constraint force profiles for pulling motion

dynamics solution are more complicated [12]. Since no friction or discontinuity is involved in our problems, our results were verified by solving the inverse dynamics problem using the joint variable profiles obtained previously. The resulting actuator torques and the constraint loads are shown to be identical to the ones obtained through our optimization algorithm. Therefore, the proposed method is proven to determine accurate joint constraint loads for a given motion of a 3-DOF planar manipulator.

Human model Consider the 21-DOF Santos$^{\text{TM}}$ digital human model of the torso and right arm (Fig. 12), which is described in detail by Kim et al. [19]. It is desired to predict the motion of pulling a lever with a constant 500-N force from an initial position $(-30, 10, 60)$ (cm) to a final position $(-30, 10, 20)$ (cm) in the global coordinate frame, where the path of the lever is given as a straight line. The time duration for the task is given as 2 seconds. For simplicity, the actuator torque limits are not taken into account in this example.

Figures 13, 14 show the predicted motions and calculated results of several notable joints. The digital human extends its torso backward to use its own body weight to sustain the large pulling force at the right hand. In Fig. 14, the large negative actuator torque values for torso axial rotations (joints 3, 6, 9, and 12) and clavicle/shoulder abductions (joints 14 and 16) for the 500-N lever-pulling indicate the major contributions of these joints to the pulling motion. These large actuator torques are used to generate the motion while resisting the large pulling force in the forward direction. In this way, the digital human can also straighten its

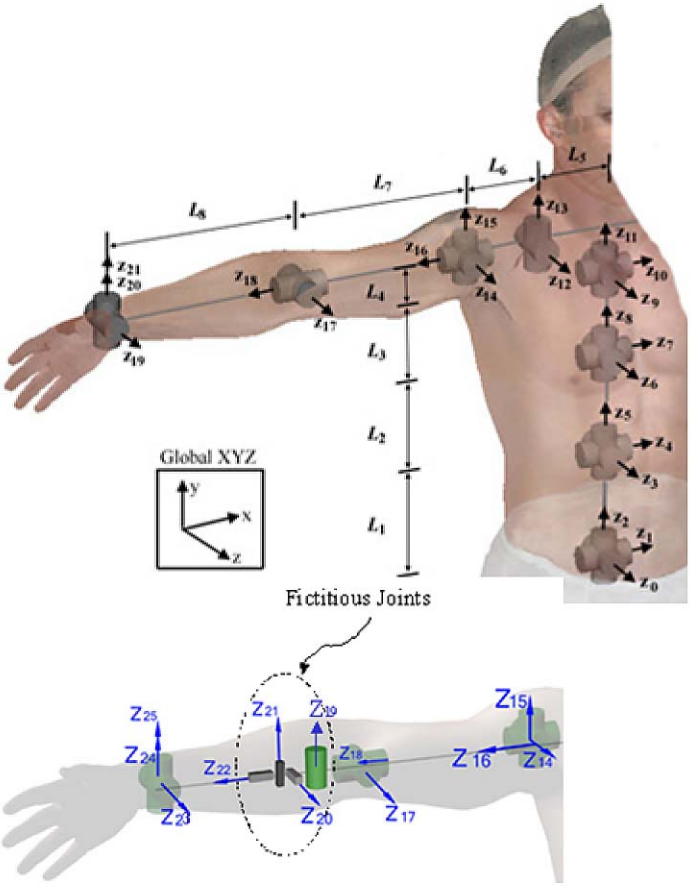
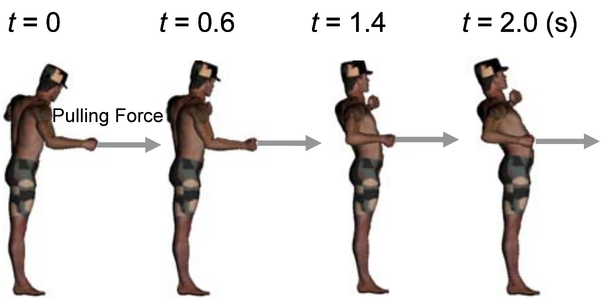


Fig. 12 A 21-DOF human model and its fictitious joints

Fig. 13 Predicted motion of 500-N one-arm lever-pulling



right arm in the initial posture to minimize the actuator torques at the wrist and elbow. This trend is similar to the previous load-effective motion results for the 3-DOF planar manipulator. These features are commonly observed in real-world human motion. When pulling or dragging a heavy object, a human usually leans his or her body in the desired direction of pulling. In other words, to accomplish a given task, humans naturally generate the effective

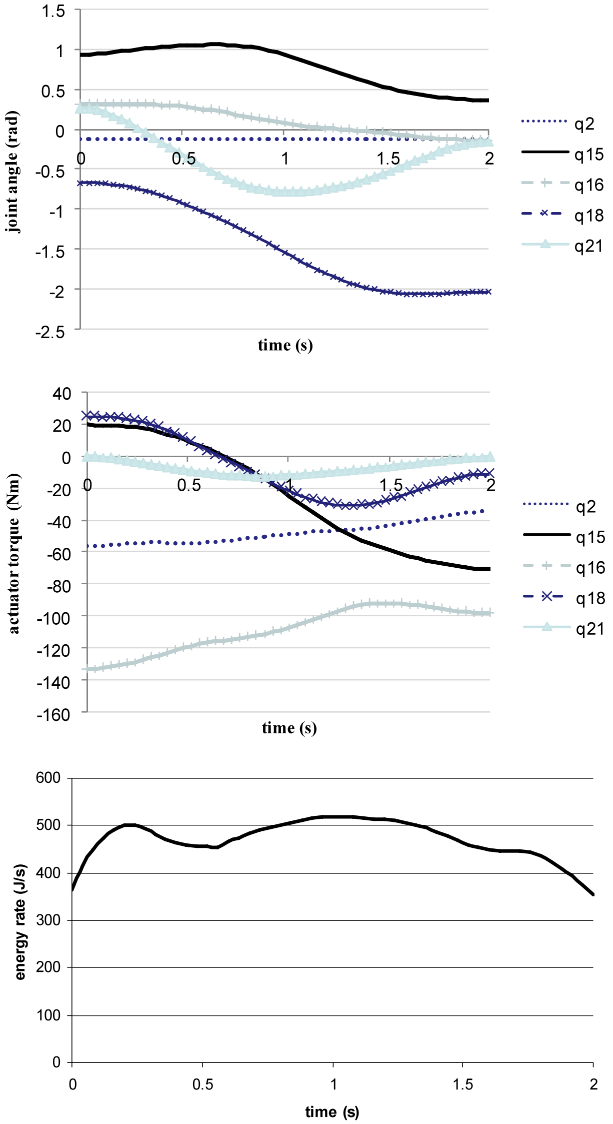


Fig. 14 Predicted joint variable, actuator torque, and energy rate profiles for lever-pulling (q_2—torso flexion/extension; q_{15}—shoulder flexion/extension; q_{16}—shoulder abduction/adduction; q_{18}—elbow flexion/extension; q_{21}—wrist flexion/extension)

motion that minimizes the required actuator torques within the actuator capacities (torque limits). A similar argument can be made for the motion of pushing a heavy object, where it is frequently observed that a human leans his or her body in the direction of pushing. The details of the results and discussion are presented by Kim et al. [19].

Next, the same task motion will be predicted along with the constraint loads at the elbow. Two revolute joints represent the motion of the elbow (flexion/extension and supination/pronation). Therefore, four fictitious joints are added to the right elbow to allow full

Table 3 Extended DH parameters for human upper body

Joint	θ	d	α	a
19	$-\pi/2+q_{19}$	0	$-\pi/2$	0
20	$-\pi/2+q_{20}$	0	$-\pi/2$	0
21	0	q_{21}	$\pi/2$	0
22	$\pi/2$	q_{22}	$\pi/2$	0
23	$\pi/2$	$24.7+q_{23}$	$\pi/2$	0
24	$\pi/2+q_{24}$	0	$\pi/2$	0
25	q_{25}	0	0	0

Table 4 Calculated joint constraint forces and moments

Fictitious joint	Due to applied force	Due to gravity	Total constraint force/moments
20	−38.63 (Nm)	−0.0226 (Nm)	−38.65 (Nm)
21	−155.67 (N)	0.0037 (N)	−155.67 (N)
22	81.50 (N)	21.24 (N)	102.74 (N)
23	−468.11 (N)	3.70 (N)	−464.41 (N)

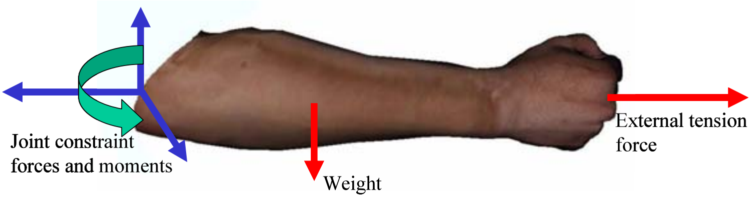


Fig. 15 Free-body diagram of the forearm and the hand with joint constraint loads

DOFs, and thus the total number of DOFs of the model is increased to 25. The fictitious joints (Fig. 12) are one revolute joint (joint 20) and three prismatic joints (joints 21, 22, and 23). The DH parameters from joint 1 to joint 18 remain the same as in the original model. The DH parameters of the modified model from joint 19 to joint 25 are given in Table 3.

The optimization results for the motion and the required actuator torques are almost identical to the previous results (again, adding the fictitious joints in the formulation did *not* affect the original optimal motion results); thus we will limit our discussion to the joint constraint loads. The calculated joint constraint forces and moments at all four fictitious joints (joints 20, 21, 22, and 23) for the motion at time 0 are listed in Table 4.

Again, the results can be roughly validated by analyzing the free-body diagram of the system (forearm and hand, Fig. 15). The calculated constraint force at joint 23 due to the applied force (−468.11 N) is very close to the amount of applied pulling force (500 N). Considering that the forearm of the motion at time 0 is almost horizontal, our result is correct (where the inertia force should be added). Also, in this configuration, the constraint force at joint 22 is almost vertical. Since the total mass of the forearm and the hand is given as 2.2 kg,

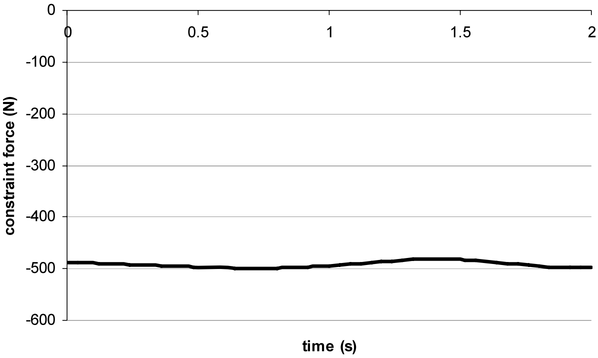


Fig. 16 Joint constraint force profile for joint 23

the weight of the forearm and hand is 21.56 N, which matches the calculated constraint force at joint 22 due to gravity (21.24 N).

The constraint force profile during the motion for joint 23 is shown in Fig. 16. Here we show only the result of joint 23 since it is the dominant constraint force among the others in terms of magnitude. Throughout the motion, the constraint force at joint 23 is maintained very close to −500 N. This is because the forearm is almost horizontal during the motion, and thus the magnitude of the constraint force at joint 23 has to be almost the same as the applied force to satisfy the force equilibrium.

For more accurate verification, our results were validated against the inverse dynamics solutions using the given joint variable profiles (as explained previously, this works for the case where no friction or discontinuity is involved). The resulting actuator torques and the constraint loads were shown to be identical to the ones obtained through our optimization algorithm, which verifies our results.

Although our example shows the determination of constraint load components at the elbow, the method can be applied to any joint of the human body, such as hips, spine, shoulders, knees, ankles, or wrists. It can also be used to determine the internal reaction forces and moments at the points that are not real joints. For instance, the reaction forces at the middle of a link can be determined by adding full six-DOF fictitious joints to the point of interest.

7 Conclusion

We proposed a method of solving for the joint constraint loads (Lagrange multipliers) for load-effective optimal dynamic motions of redundant manipulators represented in DH notation. When the DH representation is used for kinematic modeling, the relative joint constraints cannot be described explicitly using common formulation methods such as the Cartesian approach. By adding fictitious joints (thus increasing DOFs), the joint constraint loads can be obtained for the manipulators modeled in DH representation. The optimization formulation for redundant manipulator dynamic motion planning is extended to include the Lagrange multipliers associated with the fictitious joints and the corresponding zero-displacement constraints. The joint constraint loads are calculated concurrently during the process of optimal motion planning, which traditionally has been done as a post-process according to the resulting motion profiles. The results from our calculation were verified against both intuitive check and numerical inverse dynamics solutions. Furthermore, it was shown that the implementation of joint constraint loads by adding fictitious joints to the formulation do not affect the original results of the planned motion. Although the DH parameters should be modified at the specified joints, this process can easily be automated as future work. The calculated joint constraint loads can be used for stress-strain analysis of robotic systems, design of joints and links of manipulators, prediction of human injury at joints and bones, and so on. Our method will provide a viable alternative to the determination of joint constraint loads, where the equivalent DAEs are satisfied.

Acknowledgements This research is funded by the US Army Tank Automotive Research, Development, and Engineering Center (TARDEC) and the US Army Natick Soldier Systems Center.

References

1. Anand, V.B.: Computer Graphics and Geometric Modeling for Engineers. Wiley, New York (1993)
2. Arora, J.S.: Introduction to Optimum Design. McGraw-Hill, New York (1989)
3. Baysec, S., Jones, J.R.: A generalized approach for the modeling of articulated open chain planar linkages. Robotica **15**(5), 523–531 (1997)
4. Biscarini, A., Cerulli, G.: Modeling of the knee joint load in rehabilitative knee extension exercises under water. J. Biomech. **40**(2), 345–355 (2007)
5. Blajer, W.: On the determination of joint reactions in multibody mechanisms. J. Mech. Des. **126**(2), 341–350 (2004)
6. Blajer, W., Czaplicki, A.: An alternative scheme for determination of joint reaction forces in human multibody models. J. Theor. Appl. Mech. **43**(4), 813–824 (2005)
7. Brenan, K.E., Campbell, S.L., Petzold, L.R.: Numerical Solution of Initial Value Problems in Differential-Algebraic Equations. North-Holland, New York (1989)
8. Conkur, E.S., Buckingham, R.: Clarifying the definition of redundancy as used in robotics (Technical Note). Robotica **15**(5), 583–586 (1997)
9. Craig, J.J.: Introduction to Robotics: Mechanics and Control, 2nd edn. Addison-Wesley, Reading (1989)
10. Cuadrado, J., Cardenal, J., Bayo, E.: Modeling and solution methods for efficient real-time simulation of multibody dynamics. Multibody Syst. Dyn. **1**(3), 259–280 (1997)
11. Denavit, J., Hartenberg, R.S.: A kinematic notation for lower-pair mechanisms based on matrices. J. Appl. Mech. **77**, 215–221 (1955)
12. Dupont, P.E., Yamajako, S.P.: Stability of frictional contact in constrained rigid-body dynamics. IEEE Trans. Robotics Autom. **13**(2), 230–236 (1997)
13. Greenwood, D.T.: Principles of Dynamics, 2nd edn. Prentice-Hall, Englewood Cliffs (1988)
14. Hairer, E., Lubich, C., Roche, M.: The Numerical Solution of Differential-Algebraic Equations by Runge–Kutta Methods. Lecture Notes in Mathematics. Springer, Berlin (1989)
15. Haug, E.J.: Computer-Aided Kinematics and Dynamics of Mechanical Systems. Allyn and Bacon, Boston (1989)
16. Hemami, H., Wyman, B.F.: Rigid body dynamics, constraints, and inverses. J. Appl. Mech. **74**(1), 47–56 (2007)
17. Hirashima, M., Kudo, K., Ohtsuki, T.: A new non-orthogonal decomposition method to determine effective torques for three-dimensional joint rotation. J. Biomech. **40**(4), 871–882 (2007)
18. Jay, L.O.: Iterative solution of SPARK methods applied to DAEs. Numer. Algorithms **31**(1–4), 171–191 (2002)
19. Kim, J.H., Abdel-Malek, K., Yang, J., Marler, T.: Prediction and analysis of human motion dynamics performing various tasks. Int. J. Hum. Factors Model. Simul. **1**(1), 69–94 (2006)
20. Kim, J.H., Yang, J., Abdel-Malek, K.: Planning load-effective dynamic motions of redundant manipulators. In: ASME International Design Engineering Technical Conferences, 4–7 September, Las Vegas, NV (2007)
21. Langhaar, H.L.: Energy Methods in Applied Mechanics. Krieger Publishing Co. (1989)
22. McLean, S.G., Su, A., van den Bogert, A.J.: Development and validation of a 3-D model to predict knee joint loading during dynamic movement. J. Biomech. Eng. **125**(6), 864–874 (2003)
23. Negrut, D., Haug, E.J., German, H.C.: An implicit Runge–Kutta method for integration of differential-algebraic equations of multibody dynamics. Multibody Syst. Dyn. **9**(2), 121–142 (2003)
24. Nikravesh, P.E.: Computer-Aided Analysis of Mechanical Systems. Prentice-Hall, Englewood Cliffs (1988)
25. Schiehlen, W.: Multibody system dynamics: Roots and perspectives. Multibody Syst. Dyn. **1**(2), 149–188 (1997)

26. Sciavicco, L., Siciliano, B.: Modeling and Control of Robot Manipulators. McGraw-Hill, New York (1996)
27. Shabana, A.A.: Flexible multibody dynamics: Review of past and recent developments. Multibody Syst. Dyn. **1**(2), 189–222 (1997)
28. Strogatz, S.H.: Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering. Perseus Books, Cambridge (1994)