Robotics: Science and Systems 2017
Cambridge, MA, USA, July 12-16, 2017

Comprehensive Theory of Differential Kinematics and Dynamics for Motion Optimization

Ko Ayusawa and Eiichi Yoshida
CNRS-AIST JRL, UMI3218/RL, National Institute of Advanced Industrial Science and Technology (AIST), Japan.
EMail: {k.ayusawa, e.yoshida}@aist.go.jp

***Abstract*—This paper presents a novel unified theoretical framework for differential kinematics and dynamics for complex robot motion optimization. By introducing 18×18 comprehensive motion transformation matrix (CMTM), forward differential kinematics and dynamics including velocity and acceleration can be written in a simple chain product like ordinary rotational matrix. This formulation enables analytical computation of derivative of various physical quantities including joint force or torques with respect to joint coordinate variables and their derivatives for a robot trajectory in an efficient manner ($O(N_J)$, where N_J is the number of the robot's DOF), which is useful for motion optimization.**

I. Introduction

In recent robotics research, optimization has been increasing its importance in many aspects. Classic inverse kinematics and inverse dynamics problem [19] can be handled as an optimization problem of joint coordinates or torques under some constraints which are derived from physical consistency or desired tasks. Some practical optimization frameworks are proposed and applied to not only motion planning and control of a humanoid robot [25, 8], but also to motion reconstruction [26], contact estimation, and musculoskeletal analysis [7, 20] of a digital human model. Model identification problem [13] is also an optimization problem about model parameters with inverse dynamics computation. In these basic problems, only one type of physical quantities is optimized; inverse kinematics computes joint coordinates, inverse dynamics computes joint torques, identification computes inertial parameters, etc.

However, practical problems are usually their combinations and different physical quantities need to be simultaneously optimized through several time instances. Trajectory optimization under physical consistent conditions is a typical example. In locomotion planning, as violation of conditions at a certain time instance leads to future risk of falls, several sets of joint coordinates during certain period often need to be optimized simultaneously in order to predict future risks. Since this optimization usually requires huge computation cost, the balancing problem is often simplified, for example, by utilizing low-dimensional model [12]. Another example is kinematic calibration of human body segments [15] from motion capture measurement since the joint angles cannot be measured directly by encoders unlike standard robot calibration [13]. It requires simultaneous optimization of the geometric parameters and the generalized coordinates [2].

More recently, application of optimization has been intensively investigated in the field of anthropomorphic systems: humanoid robotics and human motion analysis. Humanoid robots are expected to execute more complicated and practical tasks such as disaster response [16], evaluation of human oriented products as physical human simulator [18], etc. In such applications, anthropomorphic motion optimization faces far more complicated problems combining modeling, kinematics, dynamics, planning, and control. For instance, inertial parameters identification of a humanoid robot is important to realize precise and dynamic control [3]. However, the optimal motion generation to maximize the total identification performance is a complex problem of trajectory optimization with balancing problem [6]. In the humanoid application as an active dummy for assistive device evaluation, imitation of human-like motion is necessary. This technique, called motion retargeting [10, 22], usually involves inverse kinematics problem of both human and humanoid, identification of the morphing function, and motion control of a robot with considering physical consistency. Yet another example is human simulators: recent detailed simulators are often connected to the other simulation systems like deformation computation such as FEM [1]. Those simulators are often used to optimize motion of digital human or parameters of a product to be designed with the simulator. In such problems, we usually solve the simultaneous problem of modeling, kinematics, and dynamics problems [4]. However, the above issues are currently difficult to be solved, and some of them are still open problems.

In order to establish a comprehensive optimization framework that can handle critical issues required from the computational and practical point of view, the partial derivative of any physical quantities with respect to the joint coordinates is indispensable when evaluating various types of conditions represented by the coordinates, derivatives, and forces of both Cartesian and joint space. Suleiman, et al [25] developed the fundamental framework of humanoid motion optimization. It utilizes the works of Park, et al [21] and Sohl, et al [23] and formulates the analytical partial derivative of Cartesian coordinates, their derivative, and joint torques with respect to the joint coordinates and their derivatives. In spite of its possibility of extension to handle many types of problems mentioned above, there are still two issues. First, as the formulation mainly focuses on the manifold of Cartesian spaces, it is difficult to handle a free-floating base and spherical joints that are often used in human and humanoid kinematics modeling. The second issue is the computational complexity; if we compute the partial derivative of joint torque of floating-

base side, the computational complexity is proportional to the square of the degree of freedom (DOF), i.e. O (N_J^2), where N_J is the number of DOF. It leads to huge computational cost of optimization when dealing with a large-DOF system.

In this paper, we reformulate the differential kinematics and dynamics in order to perform fast computation of the analytical partial derivative of Cartesian variables and generalized forces with respect to the joint coordinates and their derivatives. We introduce a 18-dimensional Comprehensive Motion Transformation Matrix (CMTM) in order to formulate the standard forward differential kinematics problem. This formulation makes it possible to reduce the computation of differential forward kinematics of kinematic chain to a simple chain product of the matrices in the similar manner as standard rotational matrix, or the 6 dimensional matrix used in adjoint map on SE(3) [21]. The CMTM also allows formulating an analytical form of several partial derivatives with respect to the joint coordinates and their derivatives including different types of joints. This partial derivative of link variables is extended form of basic Jacobian [14], and can be derived by the same formulation to introduce basic Jacobians. The Jacobian of joint torque is also extended form of linear/angular momentum Jacobian [11, 24], which is also formulated in the same manner thanks to the CMTM. In addition, each computational cost of new Jacobians is O(N_J).

II. Motion optimization framework

This section presents the overview of motion optimization problem and the flow of the computation. Let the generalized coordinates of a robot be $\boldsymbol{q}$, with their trajectories parameterized by $\boldsymbol{a}$ and time instance t: $\boldsymbol{q}(\boldsymbol{a},t)$. The trajectories are represented by, for example, polynomial interpolation, Fourier series, or B-splines, etc. Their derivatives $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are computed with $\boldsymbol{a}$ and t according to the implemented trajectory parameterization.

Let us concatenate $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ into $\boldsymbol{x}$, and consider physical quantities $\boldsymbol{y}$ which are represented by $\boldsymbol{x}$. The candidates of $\boldsymbol{y}$ are, for example, the position, orientation, linear and angular velocity, linear and angular acceleration of each link coordinate, the joint torques and the constraint forces acting on the joint coordinates, etc. Let $\boldsymbol{y}_{i,j}$ be i-th quantity at j-th time instance t_j, $\boldsymbol{x}_j$ be the coordinates and their derivatives at t_j, and $\boldsymbol{Y}$ are the whole set of the quantities to be evaluated. In this paper, the set of time instance t_j is given and constant, and the following optimization problem to be solved.

$$\begin{aligned} &\min_{\boldsymbol{a}} \ c(\boldsymbol{Y}) \\ &\text{subject to} \ \ \forall\, k \ \ g_k(\boldsymbol{Y}) \leq 0 \end{aligned} \tag{1}$$

where, c is the cost function to be evaluated, and g_k is k-th inequality constraint. Since equality constraints can be represented by two inequality constraints, they are summarized and represented by inequality form.

The above optimization problem is usually computationally expensive. In order to accelerate the computational speed, the efficient optimization techniques usually require analytical gradient computation of the cost function and each constraint. The gradient can be decomposed as follows:

$$\frac{\partial h}{\partial \boldsymbol{a}} = \sum_i \sum_j \frac{\partial h}{\partial \boldsymbol{y}_{i,j}} \frac{\partial \boldsymbol{y}_{i,j}}{\partial \boldsymbol{x}_j} \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{a}} \quad (h = c \ \ or \ \ g_k) \tag{2}$$

The gradient $\partial c/\partial \boldsymbol{y}_{i,j}$ is determined by the form of the evaluation function. The partial derivative $\partial \boldsymbol{x}_j/\partial \boldsymbol{a}$ can be computed from the implemented trajectory parameterization. The term $\partial \boldsymbol{y}_{i,j}/\partial \boldsymbol{x}_j$ means the partial derivative of several types of quantities of multi-body system with respect to the joint coordinates and their derivatives. The typical example is the partial derivative of the position and orientation of each link with respect to the joint coordinates which are known as the basic Jacobian [14]. In this case, the derivatives of the velocities, the accelerations, the joint torques with respect to $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ are required. By utilizing the analytical formulations for manipulators [21, 23], the motion optimization framework of **Eq.**(1) was applied for a humanoid robot by Suleiman, et al [25]. Though it can have the possibility to handle many types of motion optimization, there still remain theoretical and practical issues in order to solve the motion optimization for humanoid systems.

First, the formulation should be extended for spherical joints, free-floating base or other types of joints in order to be applied to anthropomorphic systems, because the formulations are originally for manipulators. The second issue is the computational complexity of the computation. The formulations in [25] basically utilize the classical recursive formula of forward kinematics and inverse dynamics [17]. The derivatives with respect to the coordinates of one joint are computed according to the recursive formula, and it is applied for every joint coordinates. Therefore, when computing the partial derivative of the variables of one link, the computational complexity is almost O (N_J^2), where N_J is the number of DOF.

As mentioned above, one typical example of $\partial \boldsymbol{y}_{i,j}/\partial \boldsymbol{x}_j$ is the basic Jacobian, and its computational cost of standard basic Jacobian is O (N_J). This paper introduces 18×18 matrix which can represent the forward kinematics computation including velocities and accelerations by simple chain products. The matrix has same features of 6×6 transformation matrix which represents position and orientation as mentioned later. By utilizing this matrix together with the formulation of deriving basic Jacobians, the computation method of arbitrary $\partial \boldsymbol{y}_{i,j}/\partial \boldsymbol{x}_j$ is introduced in this paper. When computing the derivatives of joint torques, the derivation form of COM Jacobians [24] are utilized. In the formulation, several types of joints also can be handled, and its computational complexity is O (N_J).

III. Mathematical notations and comprehensive motion transformation matrix

This section presents the preliminary notation of variables in the paper, and introduces a useful matrix in order to represent the forward kinematics computation including velocity and acceleration of multi-body system.

A. Definitions of basic geometric and mechanical variables

- $\boldsymbol{O}_n$ and $\boldsymbol{E}_n$ are $n \times n$ zero and identity matrix respectively
- Skew operator is represented as follows:

$$[\boldsymbol{x}\times] \triangleq \begin{bmatrix} 0 & -x_{(3)} & x_{(2)} \\ x_{(3)} & 0 & -x_{(1)} \\ -x_{(2)} & x_{(1)} & 0 \end{bmatrix} \tag{3}$$

- The position and orientation matrix of the coordinate system of a rigid body are $\boldsymbol{p}$ and $\boldsymbol{R}$ respectively.
- Let $\boldsymbol{\omega}$ and $\boldsymbol{v}$ be the linear and angular velocity represented by the local coordinate respectively, and the following relationship holds.

$$\dot{\boldsymbol{R}} = \boldsymbol{R}[\boldsymbol{\omega}\times] \tag{4}$$
$$\boldsymbol{v} = \boldsymbol{R}^T\dot{\boldsymbol{p}} \tag{5}$$

- The 6×6 transformation matrix of position $\boldsymbol{p}$ and orientation $\boldsymbol{R}$ is defined as follows:

$$\boldsymbol{A}(\boldsymbol{p},\boldsymbol{R}) \triangleq \begin{bmatrix} \boldsymbol{R} & [\boldsymbol{p}\times]\boldsymbol{R} \\ \boldsymbol{O}_3 & \boldsymbol{R} \end{bmatrix} \tag{6}$$

- Linear and angular velocities are concatenated and defined as the following vector:

$$\boldsymbol{\upsilon} \triangleq \begin{bmatrix} \boldsymbol{v}^T & \boldsymbol{\omega}^T \end{bmatrix}^T \tag{7}$$

- Let us define the operator for linear and angular velocities as follows:

$$[\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}] \triangleq \begin{bmatrix} [\boldsymbol{\omega}\times] & [\boldsymbol{v}\times] \\ \boldsymbol{O}_3 & [\boldsymbol{\omega}\times] \end{bmatrix}, \quad \boldsymbol{\upsilon}_1 \boldsymbol{\cdot} \boldsymbol{\upsilon}_2 \triangleq [\boldsymbol{\upsilon}_{1\boldsymbol{\cdot}}]\boldsymbol{\upsilon}_2 \tag{8}$$

- The above operator satisfies the binary operation axioms in Appendix A. The followings also hold:

$$\dot{\boldsymbol{A}} = \boldsymbol{A}[\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}] \tag{9}$$
$$\boldsymbol{A}[\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}]\boldsymbol{A}^{-1} = [(\boldsymbol{A}\boldsymbol{\upsilon})_{\boldsymbol{\cdot}}] \tag{10}$$

- Inertial properties of a rigid body consist of mass m, center of mass $\boldsymbol{c}$, and inertia tensor $\boldsymbol{I}_c$. They can be summarized as the following 6×6 matrix:

$$\boldsymbol{M} \triangleq \begin{bmatrix} m\boldsymbol{E}_3 & m[\boldsymbol{c}\times]^T \\ m[\boldsymbol{c}\times] & \boldsymbol{I}_c + m[\boldsymbol{c}\times][\boldsymbol{c}\times]^T \end{bmatrix} \tag{11}$$

- Let the inertial forces of a rigid body be $\widehat{\boldsymbol{f}}$ and the moment around its coordinate be $\boldsymbol{n}$. They are represented by global frame. Then, let us define 6-axis force $\boldsymbol{f}$ represented by the local coordinate as follows:

$$\boldsymbol{f} \triangleq \begin{bmatrix} \boldsymbol{R}\widehat{\boldsymbol{f}}^T & \boldsymbol{R}\boldsymbol{n}^T \end{bmatrix}^T \tag{12}$$

- The equations of motion of a rigid body are:

$$\boldsymbol{f} = \boldsymbol{M}\dot{\boldsymbol{\upsilon}} - [\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}]^T\boldsymbol{M}\boldsymbol{\upsilon} \tag{13}$$

- The variation of the equation of motion of a rigid body is written as follows by using matrix $\boldsymbol{D}$.

$$\delta\boldsymbol{f} = \boldsymbol{M}\delta\dot{\boldsymbol{\upsilon}} - \boldsymbol{D}\delta\boldsymbol{\upsilon} \tag{14}$$
$$\boldsymbol{D} \triangleq -[(\boldsymbol{M}\boldsymbol{\upsilon})_{\widehat{\boldsymbol{\cdot}}}] - [\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}]^T\boldsymbol{M} \tag{15}$$

- Operation $[\widehat{\boldsymbol{\cdot}}]$ is defined as follows:

$$[\boldsymbol{f}\widehat{\boldsymbol{\cdot}}] \triangleq \begin{bmatrix} \boldsymbol{O}_3 & \left[\widehat{\boldsymbol{f}}\times\right] \\ \left[\widehat{\boldsymbol{f}}\times\right] & [\boldsymbol{n}\times] \end{bmatrix}, \quad [\widehat{\boldsymbol{f}}_{1\boldsymbol{\cdot}}]^T\widehat{\boldsymbol{f}}_2 = [\widehat{\boldsymbol{f}}_{2}\widehat{\boldsymbol{\cdot}}]\widehat{\boldsymbol{f}}_1 \tag{16}$$

B. Comprehensive motion transformation matrix (CMTM)

Let us define the following new 18×18 matrix $\boldsymbol{X}$ and call it comprehensive motion transformation matrix (CMTM).

$$\begin{aligned} \boldsymbol{X}(\boldsymbol{A},\boldsymbol{\upsilon},\dot{\boldsymbol{\upsilon}}) &\triangleq \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{O}_6 & \boldsymbol{O}_6 \\ \boldsymbol{X}_2 & \boldsymbol{X}_1 & \boldsymbol{O}_6 \\ \boldsymbol{X}_3 & \boldsymbol{X}_2 & \boldsymbol{X}_1 \end{bmatrix} \\ &\triangleq \begin{bmatrix} \boldsymbol{A} & \boldsymbol{O}_6 & \boldsymbol{O}_6 \\ \boldsymbol{A}[\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}] & \boldsymbol{A} & \boldsymbol{O}_6 \\ \frac{1}{2}\boldsymbol{A}([\dot{\boldsymbol{\upsilon}}_{\boldsymbol{\cdot}}] + [\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}]^2) & \boldsymbol{A}[\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}] & \boldsymbol{A} \end{bmatrix} \end{aligned} \tag{17}$$

The following variation of 18 dimensional vector is defined as follows:

$$\delta\boldsymbol{x} \triangleq \begin{bmatrix} \delta\boldsymbol{\alpha}^T & \delta\boldsymbol{\upsilon}^T & \delta\dot{\boldsymbol{\upsilon}}^T \end{bmatrix}^T \tag{18}$$

where variation $\delta\boldsymbol{\alpha}$ has the following relationship.

$$[(\delta\boldsymbol{\alpha})_{\boldsymbol{\cdot}}] \triangleq \boldsymbol{A}^{-1}(\delta\boldsymbol{A}) \tag{19}$$

Vector $\delta\boldsymbol{x}$ is the concatenated vector of the variation of standard 6 dimensional coordinates, velocities, and accelerations.

In order to handle the differential operation of matrix $\boldsymbol{X}$, the following variation of 18 dimensional vector is newly defined as follows:

$$\delta\boldsymbol{\xi} \triangleq \begin{bmatrix} \delta\boldsymbol{\alpha}^T & \delta\boldsymbol{\zeta}^T & \delta\boldsymbol{\eta}^T \end{bmatrix}^T \tag{20}$$

where,

$$\delta\boldsymbol{\zeta} \triangleq \delta\boldsymbol{v} + [(\delta\boldsymbol{\alpha})_{\boldsymbol{\cdot}}\boldsymbol{v}] \tag{21}$$
$$\delta\boldsymbol{\eta} \triangleq \frac{1}{2}(\delta\dot{\boldsymbol{v}} + [(\delta\boldsymbol{\alpha})_{\boldsymbol{\cdot}}\dot{\boldsymbol{v}}] + [(\delta\boldsymbol{\zeta})_{\boldsymbol{\cdot}}\boldsymbol{v}]) \tag{22}$$

For the convenience of explanation, let us summarize the above equations as follows:

$$\delta\boldsymbol{\xi} = \boldsymbol{S}\delta\boldsymbol{x} \tag{23}$$

Matrix $\boldsymbol{S}$ transforms variation $\delta\boldsymbol{x}$ into that of that of new vector $\delta\boldsymbol{\xi}$, and is written as follows:

$$\boldsymbol{S}(\boldsymbol{\upsilon},\dot{\boldsymbol{\upsilon}}) \triangleq \begin{bmatrix} \boldsymbol{E}_6 & \boldsymbol{O}_6 & \boldsymbol{O}_6 \\ -[\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}] & \boldsymbol{E}_6 & \boldsymbol{O}_6 \\ -\frac{1}{2}([\dot{\boldsymbol{\upsilon}}_{\boldsymbol{\cdot}}] - [\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}]^2) & -\frac{1}{2}[\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}] & \frac{1}{2}\boldsymbol{E}_6 \end{bmatrix} \tag{24}$$

The inverse matrix of $\boldsymbol{S}$ always exists, and is computed as follows:

$$\boldsymbol{S}^{-1} = \begin{bmatrix} \boldsymbol{E}_6 & \boldsymbol{O}_6 & \boldsymbol{O}_6 \\ [\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}] & \boldsymbol{E}_6 & \boldsymbol{O}_6 \\ [\dot{\boldsymbol{\upsilon}}_{\boldsymbol{\cdot}}] & [\boldsymbol{\upsilon}_{\boldsymbol{\cdot}}] & 2\boldsymbol{E}_6 \end{bmatrix} \tag{25}$$

Although the variation $\delta\boldsymbol{x}$ is what we are familiar with in robotic analysis, its usage makes the forthcoming analysis of Jacobian matrices intractable. By using the newly defined variation $\delta\boldsymbol{\xi}$, the analysis becomes easier and clearer as shown later on, and once the Jacobian is derived it can be always transformed back to $\delta\boldsymbol{x}$ by the matrix $\boldsymbol{S}$.

Let us now define the following matrix and operator:

$$[\delta\boldsymbol{\xi}^{\bullet}] \triangleq \begin{bmatrix} [\delta\boldsymbol{\alpha}_{\bullet}] & \boldsymbol{O}_6 & \boldsymbol{O}_6 \\ [\delta\boldsymbol{\zeta}_{\bullet}] & [\delta\boldsymbol{\alpha}_{\bullet}] & \boldsymbol{O}_6 \\ [\delta\boldsymbol{\eta}_{\bullet}] & [\delta\boldsymbol{\zeta}_{\bullet}] & [\delta\boldsymbol{\alpha}_{\bullet}] \end{bmatrix} \quad (26)$$

$$\delta\boldsymbol{\xi}_1{}^{\bullet}\delta\boldsymbol{\xi}_2 \triangleq [\delta\boldsymbol{\xi}_1{}^{\bullet}]\delta\boldsymbol{\xi}_2 \quad (27)$$

By utilizing the above operator, the following relationship holds:

$$\delta\boldsymbol{X} = \boldsymbol{X}[(\delta\boldsymbol{\xi})^{\bullet}] \quad (28)$$

It can be check by computing each block matrix of $\delta\boldsymbol{X}_i$:

$$\begin{aligned} \delta\boldsymbol{X}_1 &= \delta\boldsymbol{A} = \boldsymbol{A}[\delta\boldsymbol{\alpha}_{\bullet}] = \boldsymbol{X}_1[\delta\boldsymbol{\alpha}_{\bullet}] && (29)\\ \delta\boldsymbol{X}_2 &= \delta\boldsymbol{A}[\boldsymbol{v}_{\bullet}] + \boldsymbol{A}[\delta\boldsymbol{v}_{\bullet}] = \boldsymbol{X}_1[\delta\boldsymbol{\zeta}_{\bullet}] + \boldsymbol{X}_2[\delta\boldsymbol{\alpha}_{\bullet}] && (30)\\ \delta\boldsymbol{X}_3 &= \frac{1}{2}\delta\boldsymbol{A}\left([\dot{\boldsymbol{v}}_{\bullet}] + [\boldsymbol{v}_{\bullet}]^2\right) \\ &\quad + \frac{1}{2}\boldsymbol{A}([\delta\dot{\boldsymbol{v}}_{\bullet}] + [\delta\boldsymbol{v}_{\bullet}][\boldsymbol{v}_{\bullet}] + [\boldsymbol{v}_{\bullet}][\delta\boldsymbol{v}_{\bullet}]) \\ &= \boldsymbol{X}_1[\delta\boldsymbol{\eta}_{\bullet}] + \boldsymbol{X}_2[\delta\boldsymbol{\zeta}_{\bullet}] + \boldsymbol{X}_3[\delta\boldsymbol{\alpha}_{\bullet}] && (31) \end{aligned}$$

Eq.(28) has the same form as **Eq.**(9). Actually, operator $[\delta\boldsymbol{\zeta}_1{}^{\bullet}\delta\boldsymbol{\zeta}_2]$ can satisfy from the binary operation axioms in Appendix A, which can be easily checked. In addition, the following equation also holds.

$$\boldsymbol{X}[\delta\boldsymbol{\zeta}^{\bullet}]\boldsymbol{X}^{-1} = [(\boldsymbol{X}\delta\boldsymbol{\zeta})^{\bullet}] \quad (32)$$

The set of matrix $\boldsymbol{X}$ and operator $(^{\bullet})$ has the similar mathematical features as matrix $\boldsymbol{A}$ and $(_{\bullet})$ (i.e. the set of Adjoint map and Lie bracket operator). It means that many formulas of kinematics operations about position and orientation can be replaced with those including velocities and accelerations. Therefore, matrix $\boldsymbol{X}$ can comprehensively handle the kinematics transformation about motion.

C. Definition and formulas of kinematics chain

This sub-section shows the notation of open kinematic chain, and important formulas about kinematics and dynamics.

- The kinematic chain is tree-structured, and the indices are chosen from the base link to the end of branches.
- $p(i)$ is the index of a root-side link connected to link i.
- $C(i)$ is the set of indices of leaves-side links connected to link i.
- $\mathscr{C}(i)$ is the set of all leaves-side links recursively connected to link i.
- $\mathscr{P}(i)$ is the set of all root-side links recursively connected to link i.
- Let us define the following sets $\hat{\mathscr{P}}(i) \triangleq \{i, \mathscr{P}(i)\}$, $\hat{\mathscr{C}}(i) \triangleq \{i, \mathscr{C}(i)\}$
- $s_{(j,i)}$ is the following selection function:

$$s_{(j,i)} = \begin{cases} 1 & (i \in \hat{\mathscr{P}}(j)) \\ 0 & (others) \end{cases} \quad (33)$$

- Let us represent quantity $\boldsymbol{y}$ of link i such as $\boldsymbol{y}_i$,
- Let us denote $\boldsymbol{y}_j^i$ as the relative variable of $\boldsymbol{y}$ from link i to j,
- $\boldsymbol{q}_i$ is n_{Ji} set of joint variables (angles), where n_j is the number of DOF of joint i, and the followings relationship holds between the joint variables and the relative velocities between link i and $p(i)$:

$$\boldsymbol{A}_i^{p(i)} = e^{[(\boldsymbol{K}_i\boldsymbol{q}_i)_{\bullet}]} \quad (34)$$

- Matrix $\boldsymbol{K}_i$ is $6 \times n_{Ji}$ constant matrix defined according to the type of joint i. For example, if joint i_1 has z-axis rotational joint and joint i_2 has a spherical joint, the corresponding matrices are followings:

$$\boldsymbol{K}_{i_1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T, \quad \boldsymbol{K}_{i_2} = \begin{bmatrix} \boldsymbol{O}_3 & \boldsymbol{E}_3 \end{bmatrix}^T, \quad (35)$$

- $\delta\boldsymbol{\theta}_i$ is a variance defined in the tangent vector space of $\boldsymbol{A}_i^{p(i)}$, which has the following differential relationship.

$$\delta\boldsymbol{A}_i^{p(i)} = \boldsymbol{A}_i^{p(i)}[\delta\boldsymbol{\alpha}_i^{p(i)}{}_{\bullet}] = \boldsymbol{A}_i^{p(i)}[(\boldsymbol{K}_i\delta\boldsymbol{\theta}_i)_{\bullet}] \quad (36)$$

  As a note, in the case of spherical joints or free floating joints, $\delta\boldsymbol{\theta}_i$ is not equal to $\delta\boldsymbol{q}_i$ because of **Eq.**(36).
- $\boldsymbol{\psi}_i$ means the joint velocity variables, and the following equations hold between $\boldsymbol{\psi}_i$ and the relative coordinates:

$$\boldsymbol{v}_i^{p(i)} = \boldsymbol{K}_i\boldsymbol{\psi}_i \quad (37)$$

- $\boldsymbol{f}_j^{p(j)}$ means the constraint force of joint j, which has the following relationship with inertial force $\boldsymbol{f}_j$ of link j and its links connected in leaves-side.

$$\boldsymbol{f}_j^{p(j)} = \boldsymbol{f}_j + \sum_{k \in C(j)} \boldsymbol{A}_k^{j\,-T}\boldsymbol{f}_k^j \quad (38)$$

  where $j = p(k)$ holds due to $k \in C(j)$.
- $\boldsymbol{\tau}_j$ represents n_{Ji} dimensional vector of joint torque, and can be extracted from 6 dimensional vector of joint constraint forces $\boldsymbol{f}_j^{p(j)}$ as follows:

$$\boldsymbol{\tau}_j = \boldsymbol{K}_j{}^T\boldsymbol{f}_j^{p(j)} \quad (39)$$

D. CMTMs in kinematic chain

Let us consider the following chain product among CMTMs:

$$\boldsymbol{X}_j = \boldsymbol{X}_i\boldsymbol{X}_j^i \quad (40)$$

The component of $\boldsymbol{X}_i$ is written down into the followings:

$$\begin{aligned} \boldsymbol{X}_{1j} &= \boldsymbol{X}_{1i}\boldsymbol{X}_{1j}^{\,i} && (41)\\ \boldsymbol{X}_{2j} &= \boldsymbol{X}_{1i}\boldsymbol{X}_{2j}^{\,i} + \boldsymbol{X}_{2i}\boldsymbol{X}_{1j}^{\,i} && (42)\\ \boldsymbol{X}_{3j} &= \boldsymbol{X}_{3i}\boldsymbol{X}_{1j}^{\,i} + \boldsymbol{X}_{2i}\boldsymbol{X}_{2j}^{\,i} + \boldsymbol{X}_{1i}\boldsymbol{X}_{3j}^{\,i} && (43) \end{aligned}$$

Now let us see **Eq.**(41), **Eq.**(42), and **Eq.**(43) respectively. From **Eq.**(17), **Eq.**(41) can be transformed into:

$$\boldsymbol{A}_i = \boldsymbol{A}_i\boldsymbol{A}_j^i \quad (44)$$

Therefore, **Eq.**(41) means the standard forward kinematics operation.

Next, **Eq.**(42) can be converted into:

$$\begin{aligned} \boldsymbol{A}_j[\boldsymbol{v}_{j\bullet}] &= \boldsymbol{A}_i\boldsymbol{A}_j^i[\boldsymbol{v}_{j\bullet}^i] + \boldsymbol{A}_i[\boldsymbol{v}_{i\bullet}]\boldsymbol{A}_j^i \\ &= \boldsymbol{A}_j[(\boldsymbol{v}_j^i + \boldsymbol{A}_j^{i\,-1}\boldsymbol{v}_i)_{\bullet}] && (45) \end{aligned}$$

From the above equation, the followings holds identically.

$$\boldsymbol{v}_j = \boldsymbol{v}_j^i + \boldsymbol{A}_j^{i\,-1}\boldsymbol{v}_i \tag{46}$$

where, **Eq.**(46) indicates that **Eq.**(42) means the differential forward kinematics operation about linear and angular velocities.

Finally, **Eq.**(43) can be written down as follows:

$$\begin{aligned} & \frac{1}{2}\boldsymbol{A}_j\left([\dot{\boldsymbol{v}}_j\boldsymbol{.}]+[\boldsymbol{v}_j\boldsymbol{.}]^2\right) \\ = \; & \frac{1}{2}\boldsymbol{A}_i\left([\dot{\boldsymbol{v}}_i\boldsymbol{.}]+[\boldsymbol{v}_i\boldsymbol{.}]^2\right)\boldsymbol{A}_j^i + \boldsymbol{A}_i[\boldsymbol{v}_i\boldsymbol{.}]\boldsymbol{A}_j^i[\boldsymbol{v}_j^i\boldsymbol{.}] \\ & + \frac{1}{2}\boldsymbol{A}_i\boldsymbol{A}_j^i\left([\dot{\boldsymbol{v}}_j^i\boldsymbol{.}]+[\boldsymbol{v}_j^i\boldsymbol{.}]^2\right) \\ = \; & \frac{1}{2}\boldsymbol{A}_j\left([(\dot{\boldsymbol{v}}_j^i + \boldsymbol{A}_j^{i\,-1}\dot{\boldsymbol{v}}_i + (\boldsymbol{A}_j^{i\,-1}\boldsymbol{v}_i)\boldsymbol{.}\,\boldsymbol{v}_j^i)\boldsymbol{.}]+[\boldsymbol{v}_j\boldsymbol{.}]^2\right) \end{aligned} \tag{47}$$

Therefore, the followings holds identically.

$$\dot{\boldsymbol{v}}_j = \dot{\boldsymbol{v}}_j^i + \boldsymbol{A}_j^{i\,-1}\dot{\boldsymbol{v}}_i + (\boldsymbol{A}_j^{i\,-1}\boldsymbol{v}_i)\boldsymbol{.}\,\boldsymbol{v}_j^i \tag{48}$$

where, **Eq.**(48) implies that **Eq.**(43) is equivalent with the differential forward kinematics operation about linear and angular accelerations.

As can be seen from all the above, the chain products of CMTMs in **Eq.**(40) represent the standard and differential kinematics computation of a kinematic chain. The formulation using CMTM can therefore handle the kinematics operations in a comprehensive manner. This feature becomes a strong advantage when introducing arbitrary Jacobian matrices in the next section.

Now let us define the following variation of $3n_{Ji}$ dimensional vector consisting of the variations of joint variable, velocity, and its derivative:

$$\delta\boldsymbol{\chi} \;\triangleq\; \left[\delta\boldsymbol{\theta}^T \quad \delta\boldsymbol{\psi}^T \quad \delta\dot{\boldsymbol{\psi}}^T\right]^T \tag{49}$$

According to **Eq.**(20), **Eq.**(36), and **Eq.**(37), the relationship between $\delta\boldsymbol{x}_i^{p(i)}$ and $\delta\boldsymbol{\chi}_i$ is summarized as follows:

$$\delta\boldsymbol{x}_i^{p(i)} = \boldsymbol{G}_i\delta\boldsymbol{\chi}_i \triangleq \begin{bmatrix} \boldsymbol{K}_i & \boldsymbol{O}_6 & \boldsymbol{O}_6 \\ \boldsymbol{O}_6 & \boldsymbol{K}_i & \boldsymbol{O}_6 \\ \boldsymbol{O}_6 & \boldsymbol{O}_6 & \boldsymbol{K}_i \end{bmatrix}\delta\boldsymbol{\chi}_i \tag{50}$$

The relationship among the variables of link and joint coordinates and their variations is summarized in **Fig.**1.

IV. Computation of arbitrary Jacobians

This section shows the different types of arbitrary partial derivatives $\partial\boldsymbol{y}_{i,j}/\partial\boldsymbol{x}_j$ used in **Eq.**(2) by utilizing CMTM. Though $\partial\boldsymbol{y}_{i,j}/\partial\boldsymbol{x}_j$ are strictly speaking not true Jacobian matrices as in the case of basic Jacobians, however, in this paper, let us call them Jacobians for descriptive purposes.

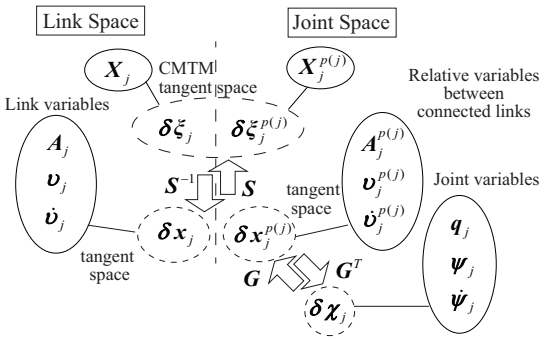

Fig. 1. Overview of the relationship among the varables used in the paper

A. Jacobians of link posture, velocity and acceleration

Let us compute matrix $\boldsymbol{J}_j$ that converts variation $\delta\boldsymbol{\chi}_{all}$ of all joints to variation $\delta\boldsymbol{x}_j$ of link j as follows:

$$\delta\boldsymbol{x}_j = \boldsymbol{J}_j\delta\boldsymbol{\chi}_{\text{all}} = \sum_k \boldsymbol{J}_{(j,k)}\delta\boldsymbol{\chi}_k \tag{51}$$

As mentioned in the previous section, matrix $\boldsymbol{X}$ has the same features as $\boldsymbol{A}$, and matrix $\boldsymbol{J}_j$ can be computed in a similar manner when computing standard basic Jacobian matrix.

Now let us consider $\boldsymbol{X}_j$ of link j. The following chain products hold among CMTMs:

$$\boldsymbol{X}_j = \boldsymbol{X}_{p(k)}\boldsymbol{X}_k^{p(k)}\boldsymbol{X}_j^k \tag{52}$$

The variation of $\boldsymbol{X}_j$ can be computed according to chain products **Eq.**(52):

$$\delta\boldsymbol{X}_j = \sum_{k\in\hat{\mathscr{P}}(j)} \boldsymbol{X}_{p(k)}\delta\boldsymbol{X}_k^{p(k)}\boldsymbol{X}_j^k \tag{53}$$

By utilizing **Eq.**(28) and **Eq.**(32), the above equation can be transformed into:

$$\begin{aligned} \boldsymbol{X}_j[(\delta\boldsymbol{\xi}_j)^\bullet] &= \sum_{k\in\hat{\mathscr{P}}(j)} \boldsymbol{X}_{p(k)}\boldsymbol{X}_k^{p(k)}[(\delta\boldsymbol{\xi}_k^{p(k)})^\bullet]\boldsymbol{X}_j^k \\ &= \boldsymbol{X}_j\sum_{k\in\hat{\mathscr{P}}(j)}[(\boldsymbol{X}_k^j\;\delta\boldsymbol{\xi}_k^{p(k)})^\bullet] \end{aligned} \tag{54}$$

According to the above equation, the following equation holds identically.

$$\delta\boldsymbol{\xi}_j = \sum_{k\in\hat{\mathscr{P}}(j)} \boldsymbol{X}_k^j\;\delta\boldsymbol{\xi}_k^{p(k)} \tag{55}$$

The coefficient matrix of **Eq.**(55) means the Jacobian matrix with respect to $\delta\boldsymbol{\xi}$, and each block matrix is equal to relative CMTM $\boldsymbol{\xi}_k^{p(k)}$.

Since the desired Jacobian matrix is the one with respect to $\delta\boldsymbol{x}_j$, let us compute it by transforming variations of **Eq.**(55) from $\delta\boldsymbol{\xi}$ to $\delta\boldsymbol{x}$ and from $\delta\boldsymbol{x}$ to $\delta\boldsymbol{\chi}$.

First, by utilizing **Eq.**(23), the following equation is obtained.

$$\delta\boldsymbol{x}_j = \sum_{k\in\hat{\mathscr{P}}(j)} \widehat{\boldsymbol{X}}_k^j\delta\boldsymbol{x}_k^{p(k)} \tag{56}$$

$$\widehat{\boldsymbol{X}}_k^j \;\triangleq\; \boldsymbol{S}_j{}^{-1}\boldsymbol{X}_k^j\boldsymbol{S}_k^{p(k)} \tag{57}$$

The next transformation can be performed with **Eq.**(50) as follows:

$$\delta \boldsymbol{x}_j = \sum_k \boldsymbol{J}_{(j,k)} \delta \boldsymbol{\chi}_k^{p(k)} \quad (58)$$

$$\boldsymbol{J}_{(j,k)} \triangleq s_{(j,k)} \widehat{\boldsymbol{X}}_k^j \boldsymbol{G}_k \quad (59)$$

From the above, the Jacobian matrix shown in **Eq.**(51) was finally derived as **Eq.**(59). Since the direct computation of 18×18 matrix products in **Eq.**(57) is computationally inefficient, the solution of 6×6 block matrices in $\widehat{\boldsymbol{X}}_j^i$ is written down in Appendix B.

B. Jacobians of link inertial forces

This subsection derives matrix $\boldsymbol{L}_j$ which converts variation $\delta \boldsymbol{\chi}_{all}$ of all joints to force variation $\boldsymbol{f}_j$ of link j as follows:

$$\delta \boldsymbol{f}_j = \boldsymbol{L}_j \delta \boldsymbol{\chi}_{\text{all}} = \sum_k \boldsymbol{L}_{(j,k)} \delta \boldsymbol{\chi}_k \quad (60)$$

Let us first consider the variation of equations of motion of link j. According to **Eq.**(14), the following equations are obtained:

$$\delta \boldsymbol{f}_j = \boldsymbol{H}_j \delta \boldsymbol{x}_j \quad (61)$$

$$\boldsymbol{H}_j \triangleq \begin{bmatrix} \boldsymbol{O}_6 & \boldsymbol{D}_j & \boldsymbol{M}_j \end{bmatrix} \quad (62)$$

From **Eq.**(61) and **Eq.**(51), variation $\delta \boldsymbol{x}_j$ can be transformed into $\boldsymbol{J}_{(j,k)}$. The above equation can be also transformed into the followings:

$$\delta \boldsymbol{f}_j = \boldsymbol{H}_j \sum_i \boldsymbol{J}_{(j,k)} \delta \boldsymbol{\chi}_k \quad (63)$$

Therefore, the Jacobian matrix shown in **Eq.**(60) could be derived as followings:

$$\boldsymbol{L}_{(j,k)} = \boldsymbol{H}_j \boldsymbol{J}_{(j,k)} \quad (64)$$

For the sake of forthcoming discussions, let us transform **Eq.**(61) to the linear form with respect to $\delta \boldsymbol{\xi}_j$ by using **Eq.**(23).

$$\delta \boldsymbol{f}_j = \boldsymbol{H}_j {\boldsymbol{S}_j}^{-1} \delta \boldsymbol{\xi}_j \quad (65)$$

C. Jacobians of joint constraint forces

In this subsection, we compute matrix $\boldsymbol{N}_j$ which converts variation $\delta \boldsymbol{\chi}_{all}$ of all joints to force variation $\boldsymbol{f}_j^{p(j)}$ of link j as follows:

$$\delta \boldsymbol{f}_j^{p(j)} = \boldsymbol{N}_j \delta \boldsymbol{\chi}_{\text{all}} = \sum_k \boldsymbol{L}_{(j,k)} \delta \boldsymbol{\chi}_k \quad (66)$$

From **Eq.**(38), the following equation about joint constraint force $\boldsymbol{f}_j^{p(j)}$ can be obtained.

$$\delta \boldsymbol{f}_j^{p(j)} = \delta \boldsymbol{f}_j + \sum_{k \in C(j)} ({\boldsymbol{A}_k^{p(k)}}^{-T} \delta \boldsymbol{f}_k^{p(k)} + \delta {\boldsymbol{A}_k^{p(k)}}^{-T} \boldsymbol{f}_k^{p(k)}) \quad (67)$$

According to **Eq.**(65), the above equation can be transformed into:

$$\delta \boldsymbol{f}_j^{p(j)} = \left(\boldsymbol{H}_j {\boldsymbol{S}_j}^{-1}\right) \delta \boldsymbol{\xi}_j + \sum_{k \in C(j)} ({\boldsymbol{A}_k^{p(k)}}^{-T} \delta \boldsymbol{f}_k^{p(k)} + \delta {\boldsymbol{A}_k^{p(k)}}^{-T} \boldsymbol{f}_k^{p(k)}) \quad (68)$$

On the other hand, the following equation holds from **Eq.**(55).

$$\begin{aligned} \delta \boldsymbol{\xi}_j &= \boldsymbol{X}_{p(j)}^j \left(\sum_{l \in \hat{\mathscr{P}}(p(j))} \boldsymbol{X}_l^{p(j)} \, \delta \boldsymbol{\xi}_l^{p(l)} \right) + \delta \boldsymbol{\xi}_j^{p(j)} \\ &= \boldsymbol{X}_{p(j)}^j \delta \boldsymbol{\xi}_{p(j)} + \delta \boldsymbol{\xi}_j^{p(j)} \end{aligned} \quad (69)$$

Since $\boldsymbol{A}^{-T}$ and $\boldsymbol{X}$ are transformation matrices, **Eq.**(69) and **Eq.**(68) have the same form of **Eq.**(90) and **Eq.**(90) in Appendix C respectively. According to the similar derivations of **Eq.**(92), **Eq.**(94), **Eq.**(95) in Appendix C, the following recursive formula can be obtained.

$$\delta \boldsymbol{f}_j^{p(j)} = \widehat{\boldsymbol{H}}_j \delta \boldsymbol{x}_j + \widehat{\boldsymbol{h}}_j (= \widehat{\boldsymbol{H}}_j {\boldsymbol{S}_j}^{-1} \delta \boldsymbol{\xi}_j + \widehat{\boldsymbol{h}}_j) \quad (70)$$

where,

$$\widehat{\boldsymbol{H}}_j = \boldsymbol{H}_j + \sum_{k \in C(j)} {\boldsymbol{A}_k^j}^{-T} \widehat{\boldsymbol{H}}_k {\boldsymbol{S}_k}^{-1} \boldsymbol{X}_j^k \boldsymbol{S}_j \quad (71)$$

$$\widehat{\boldsymbol{h}}_j = \sum_{k \in C(j)} \left({\boldsymbol{A}_k^j}^{-T} \widehat{\boldsymbol{H}}_k {\boldsymbol{S}_k}^{-1} \delta \boldsymbol{\xi}_k^{p(k)} + \delta {\boldsymbol{A}_k^{p(k)}}^{-T} \boldsymbol{f}_k^{p(k)} + {\boldsymbol{A}_k^j}^{-T} \widehat{\boldsymbol{h}}_k \right) \quad (72)$$

It should be noted that, if the set of $\boldsymbol{A}^{-T}$ and $\boldsymbol{X}$ is replaced with $\boldsymbol{A}^{-T}$ and $\boldsymbol{A}$, and if there exists no bias terms like $\delta \boldsymbol{\xi}_j^{p(j)}$, the transformation based on Appendix C shows the same formula when introducing linear and angular momentum Jacobian.

Let us expand the term of ${\boldsymbol{A}_k^j}^{-T} \widehat{\boldsymbol{h}}_k$ in **Eq.**(72) by its recursive computation toward leaves-link side. In addition, by the transformation from $\delta \boldsymbol{\xi}$ to $\delta \boldsymbol{x}$ with **Eq.**(23), **Eq.**(72) can be transformed into:

$$\widehat{\boldsymbol{h}}_j = \sum_{k \in \mathscr{C}(j)} \left({\boldsymbol{A}_k^j}^{-T} \widehat{\boldsymbol{H}}_k {\boldsymbol{S}_k}^{-1} \boldsymbol{S}_k^{p(k)} \delta \boldsymbol{x}_k^{p(k)} + {\boldsymbol{A}_{p(k)}^j}^{-T} \delta {\boldsymbol{A}_k^{p(k)}}^{-T} \boldsymbol{f}_k^{p(k)} \right) \quad (73)$$

There also exists the following conversion of variation $\delta \boldsymbol{A}_k^{p(k)-T}$ as follows:

$$\begin{aligned} \delta {\boldsymbol{A}_k^{p(k)}}^{-T} \boldsymbol{f}_k^{p(k)} &= -{\boldsymbol{A}_k^{p(k)}}^{-T} [\delta \boldsymbol{\alpha}_k^{p(k)} \boldsymbol{.}]^T \boldsymbol{f}_k^{p(k)} \\ &= -{\boldsymbol{A}_k^{p(k)}}^{-T} [\boldsymbol{f}_k^{p(k)} \boldsymbol{:}] \delta \boldsymbol{\alpha}_k^{p(k)} \end{aligned} \quad (74)$$

According to **Eq.**(74) and equation $\widehat{\boldsymbol{X}}_{(k,k)} = {\boldsymbol{S}_k}^{-1} \boldsymbol{S}_k^{p(k)}$ (which is from **Eq.**(57)), **Eq.**(73) can be converted to:

$$\widehat{\boldsymbol{h}}_j = \sum_{k \in \mathscr{C}(j)} \left({\boldsymbol{A}_k^j}^{-T} \widehat{\boldsymbol{H}}_k \widehat{\boldsymbol{X}}_{(k,k)} - {\boldsymbol{A}_k^j}^{-T} [\boldsymbol{f}_k^{p(k)} \boldsymbol{:}] \boldsymbol{T} \right) \delta \boldsymbol{x}_k^{p(k)} \quad (75)$$

where,

$$\boldsymbol{T} \triangleq \begin{bmatrix} \boldsymbol{E}_6 & \boldsymbol{O}_6 & \boldsymbol{O}_6 \end{bmatrix} \quad (76)$$

By substituting **Eq.**(71) and **Eq.**(75) for **Eq.**(70), and with converting the variations from $\delta \boldsymbol{x}_k^{p(k)}$ to $\delta \boldsymbol{\chi}_k^{p(k)}$, the following

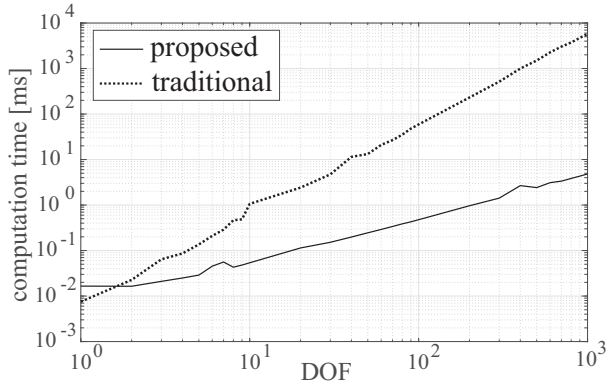

Fig. 2. Comparison of computation time of joint torque Jacobian matrix.

equation holds.

$$\delta \boldsymbol{f}_j^{p(j)} = \sum_{k \in \hat{\mathscr{P}}(j)} \widehat{\boldsymbol{H}}_j \boldsymbol{J}_{(j,k)} \delta \boldsymbol{\chi}_k^{p(k)} + \sum_{k \in \mathscr{C}(j)} \boldsymbol{A}_k^{j-T} \left(\widehat{\boldsymbol{H}}_k \boldsymbol{J}_{(k,k)} - [\boldsymbol{f}_k^{p(k)} \hat{\cdot}] \boldsymbol{T} \boldsymbol{G}_k \right) \delta \boldsymbol{\chi}_k^{p(k)} \quad (77)$$

Given all the above, the Jacobian matrix shown in **Eq.**(66) could be finally shown as followings:

$$\boldsymbol{N}_{(j,k)} = \begin{cases} \widehat{\boldsymbol{H}}_j \boldsymbol{J}_{(j,k)} & (k \in \hat{\mathscr{P}}(j)) \\ \boldsymbol{A}_k^{j-T} \left(\widehat{\boldsymbol{H}}_k \boldsymbol{J}_{(k,k)} - [\boldsymbol{f}_k^{p(k)} \hat{\cdot}] \boldsymbol{T} \boldsymbol{G}_k \right) & (k \in \mathscr{C}(j)) \\ \boldsymbol{O}_{6 \times 18} & (others) \end{cases} \quad (78)$$

The computation of **Eq.**(78) requires matrix $\widehat{\boldsymbol{H}}_j$. It means that $\widehat{\boldsymbol{H}}_j$ for all link j needs to be computed in advance according to recursive formula **Eq.**(71). After updating $\widehat{\boldsymbol{H}}_j$, matrix $\boldsymbol{N}_{(j,k)}$ for any j and k can be directly computed by **Eq.**(78). The direct computation of **Eq.**(71) with 18×18 matrices is computationally inefficient. The final form of $\widehat{\boldsymbol{H}}_j$ derived from **Eq.**(71) is written down in Appendix D.

V. Numerical examples

This section shows the comparison of computation time of Jacobian matrices between the two approaches: the proposed method and the traditional method in [25]. Since the formulations in [25] only handle 1-DOF joints, it was tested by using the serial manipulator which has N rotational joints. The Jacobian matrix of the joint torque of the first rotational joint was computed by changing the number of joints. It was tested by the computer with Intel(R) Xeon(R) CPU E3-1535M v5. The proposed method, of course, could generate the same Jacobian matrices as those from the classical method. **Fig.**2 shows the results of the computational time of the two methods. The computational complexity of the conventional method was $O(N^2)$; on the other hand, the proposed method showed $O(N)$, and the computational time was significantly improved in the large-DOF cases.

This section also validates the Jacobian in the case of spherical joints; we tested a simple example of motion optimization by using a serial manipulator with 5 spherical joints, as shown in **Fig.**3. This test assumes no gravity. Each segmented link has save physical quantities: $\boldsymbol{p}_i^{p(i)} = [0\ 0\ 0.2]$ [m], $m_i = 3.0$ [kg], $\boldsymbol{c}_i = [0\ 0\ 0.1]$ [kgm], $\boldsymbol{I}_i = \mathrm{diag}([11.2\ 11.2\ 2.4]) * 10^{-3}$ [kgm^2]. Let $\boldsymbol{p}_e^{(t)}$ be the position of the end-effector at time instance t, and $\tau_{j,k} (1 \le j \le 5, 1 \le k \le 3)$ be k-th component of torques of j-th spherical joint.

In this planning, the total time length of the trajectory is assumed to be 2 [s], and its sampling rate is 1 [ms]. The variables of the spherical joints are represented by angle-axis vector, and their trajectories was modeled by using B-splines as shown in [25]. The number of B-spline basis is 24 and the timespan between two bases is 0.1 [s]. The position of the end-effector starts from $\boldsymbol{p}_e^0 = [0\ 0\ 1]$ with zero joint velocities and accelerations, then passes through $\boldsymbol{p}_e^1 = [0\ 1\ 0]$ at $t = 1.0$ [s], and stops at $\boldsymbol{p}_e^2 = [1\ 0\ 0]$ with zero joint velocities and accelerations. During the movement, the joint torque limitations are considered; $|\tau_{1,k}| < 0.1$ [N] and $|\tau_{j,k}| < 20.0$ [N] $(2 \le j \le 5)$. In order to avoid self-collision, the distance constraints are also assumed; $||\boldsymbol{p}_k - \boldsymbol{p}_l|| \ge 0.1 (k \ne l)$. All the constraint conditions were converted to the penalty cost functions according to penalty function method. The optimization itself was solved by quasi-Newton method [9] by utilizing the proposed method. The generated trajectory is shown in **Fig.**3. The end-effector successfully passed through the targeted positions without self collision. The result of joint torque trajectories of first joint is shown in **Fig.**4, which all satisfy the joint torque limitations. The other joint torques also satisfied the limitations.

Motion optimization framework is useful to solve several practical applications as mentioned in the introduction. More complicated cases as shown in [5] will be investigated in future works.

VI. Conclusion

This paper presents the comprehensive theory of differential kinematics and dynamics in order to derive the analytical partial derivative of both link/joint quantities with respect to joint coordinates and their derivatives. First, the 18×18 comprehensive motion transformation matrix (CMTM) and 18 dimensional product operation is introduced for comprehensive kinematics formulation, which allows a simple chain product of the matrices to represent the standard and differential forward kinematics of a kinematics chain. They also have the same features as the rotation matrix and the 6×6 transformation matrix of screws, and their product operations. By utilizing CMTM, the partial derivative of link coordinates and their derivatives were derived in the same manner as when introducing the basic Jacobians by just replacing the 6×6 transformation matrices with CMTM in their formulations.

This novel theoretical framework brought the following contributions with respect to related work. The partial derivatives of each generalized force with respect to joint coordinates and their derivatives were also demonstrated. The procedure of derivation also has similarity with that of linear and angular momentum Jacobians. By utilizing the CMTM, each new Jacobians could be computed by O(N_J). The formulation can also handle different types of joints like spherical joints or free-floating base.

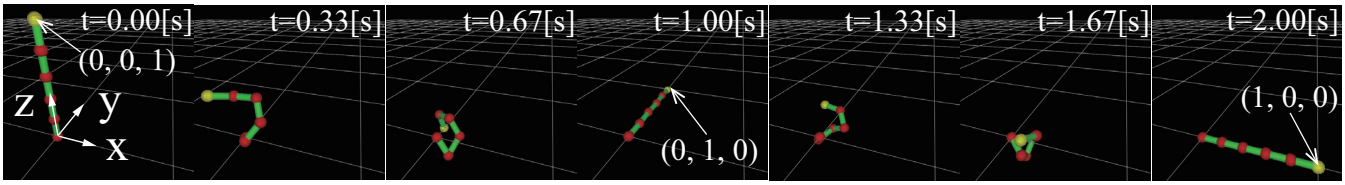

Fig. 3. Snapshots of the generated motion of the serial manipulator. The end-effector could go through the target points; it started from [0 0 1], passed through [0 1 0] at 1.0 [s], and stopped at [1 0 0]. All the other constraints like self-collision could also be archived.

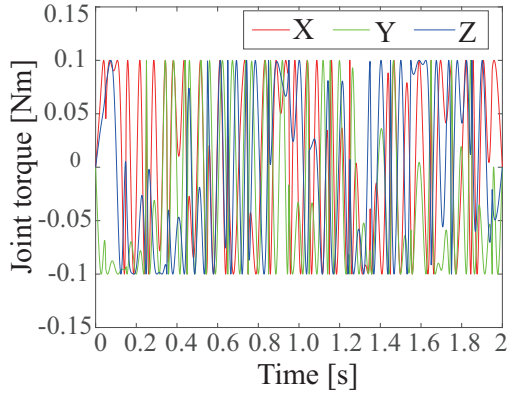

Fig. 4. Joint torque trajectories of the first spherical joint.

This work was partly supported by JSPS Grant-in-Aid for Scientific Research (A) Number 17H00768.

APPENDIX

A. Binary operation axioms

The bilinearity, alternativity, Jacobi identity axioms and anticommutativity with a binary operation $[,]$ are introduced:

$$[(a\boldsymbol{x}_1+b\boldsymbol{x}_2),\boldsymbol{x}_3]=a[\boldsymbol{x}_1,\boldsymbol{x}_3]+b[\boldsymbol{x}_2,\boldsymbol{x}_3] \tag{79}$$

$$[\boldsymbol{x}_3,(a\boldsymbol{x}_1+b\boldsymbol{x}_2)]=a[\boldsymbol{x}_3,\boldsymbol{x}_1]+b[\boldsymbol{x}_3,\boldsymbol{x}_2] \tag{80}$$

$$[\boldsymbol{x}_1,\boldsymbol{x}_1]=\boldsymbol{0} \tag{81}$$

$$[\boldsymbol{x}_1,[\boldsymbol{x}_2,\boldsymbol{x}_3]]+[\boldsymbol{x}_1,[\boldsymbol{x}_2,\boldsymbol{x}_3]]+[\boldsymbol{x}_1,[\boldsymbol{x}_2,\boldsymbol{x}_3]]=\boldsymbol{0} \tag{82}$$

$$[\boldsymbol{x}_1,\boldsymbol{x}_2]=-[\boldsymbol{x}_2,\boldsymbol{x}_1] \tag{83}$$

(In this paper, $[,]$ corresponds with $[\times]$, $[\boldsymbol{.}]$, or $[^{\bullet}]$.)

B. Structure of 18×18 matrix $\hat{\mathbf{X}}_k^j$ in Eq.(57)

Here are 6×6 block matrices in $\widehat{\boldsymbol{X}}_k^j$ of **Eq.**(57), when writing down with the notation of $\widehat{\boldsymbol{X}}_j^i$.

$$\widehat{\boldsymbol{X}}_k^j \triangleq \begin{bmatrix} \boldsymbol{A}_j^{k\,-1} & \boldsymbol{O}_6 & \boldsymbol{O}_6 \\ \widehat{\boldsymbol{X}}_{(j,k)}^{(2)} & \boldsymbol{A}_j^{k\,-1} & \boldsymbol{O}_6 \\ \widehat{\boldsymbol{X}}_{(j,k)}^{(4)} & \widehat{\boldsymbol{X}}_{(j,k)}^{(3)} & \boldsymbol{A}_j^{k\,-1} \end{bmatrix} \tag{84}$$

$$\widehat{\boldsymbol{X}}_{(j,k)}^{(2)} = \boldsymbol{A}_j^{k\,-1}[\widehat{\boldsymbol{v}}_k^{p(k)}\,\boldsymbol{.}\,] \tag{85}$$

$$\widehat{\boldsymbol{X}}_{(j,k)}^{(3)} = \boldsymbol{A}_j^{k\,-1}[(\widehat{\boldsymbol{v}}_k^{p(k)}-\boldsymbol{A}_j^k\boldsymbol{v}_j^k)\,\boldsymbol{.}\,] \tag{86}$$

$$\widehat{\boldsymbol{X}}_{(j,k)}^{(4)} = \boldsymbol{A}_j^{k\,-1}([\widehat{\boldsymbol{v}}_k^{p(k)}\,\boldsymbol{.}\,]-[(\boldsymbol{v}_k^{p(k)}+\boldsymbol{A}_j^k\boldsymbol{v}_j^k)\,\boldsymbol{.}\,][\widehat{\boldsymbol{v}}_k^{p(k)}\,\boldsymbol{.}\,]) \tag{87}$$

$$\widehat{\boldsymbol{v}}_j^k \triangleq \boldsymbol{v}_j-\boldsymbol{v}_j^k \tag{88}$$

$$\widehat{\dot{\boldsymbol{v}}}_j^k \triangleq \dot{\boldsymbol{v}}_j-\dot{\boldsymbol{v}}_j^k-[\widehat{\boldsymbol{v}}_j^k\boldsymbol{.}\,\boldsymbol{v}_j^k] \tag{89}$$

C. Recursive formulas of kinematic chain

Let $\boldsymbol{S}_j^i$ an arbitrary transformation matrix, which is non-singular and satisfies $\boldsymbol{S}_k^i=\boldsymbol{S}_j^i\boldsymbol{S}_k^j$. In regards to $\boldsymbol{S}_j^i$, physical quantity $\boldsymbol{a}_j\in\mathbb{R}^m$ has the following recursive formulas:

$$\boldsymbol{a}_j=\boldsymbol{S}_i^j\boldsymbol{a}_i+\boldsymbol{c}_j \tag{90}$$

where, $\boldsymbol{c}_j$ is a bias term. Similarly, let us consider another transformation matrix $\boldsymbol{U}_j^i$ and physical quantity $\boldsymbol{b}_i\in\mathbb{R}^n$, there also exists the following recursive formulas.

$$\boldsymbol{b}_i=\boldsymbol{P}_i\boldsymbol{V}_i\boldsymbol{a}_i+\boldsymbol{d}_i+\sum_{j\in C(i)}\boldsymbol{U}_i^j\boldsymbol{b}_i \tag{91}$$

where, $\boldsymbol{d}_j$ is bias terms, $\boldsymbol{V}_i\in\mathbb{R}^{m\times m}$ is an arbitrary non-singular matrix, and $\boldsymbol{P}_i\in\mathbb{R}^{n\times m}$ maps the space of $\boldsymbol{d}_i$ into $\boldsymbol{c}_i$.

Then, let us assume that **Eq.**(91) has the following form:

$$\boldsymbol{b}_i=\widehat{\boldsymbol{P}}_i\boldsymbol{V}_i\boldsymbol{a}_i+\widehat{\boldsymbol{d}}_i \tag{92}$$

By substituting **Eq.**(90) and **Eq.**(92), **Eq.**(91) is to be:

$$\boldsymbol{b}_i=\left(\boldsymbol{P}_i+\sum_{j\in C(i)}\boldsymbol{U}_j^i\widehat{\boldsymbol{P}}_j\boldsymbol{V}_j\boldsymbol{S}_i^j{\boldsymbol{V}_i}^{-1}\right)\boldsymbol{V}_i\boldsymbol{a}_i +\boldsymbol{d}_i+\sum_{j\in C(i)}\boldsymbol{U}_j^i(\widehat{\boldsymbol{P}}_j\boldsymbol{V}_i\boldsymbol{c}_j+\widehat{\boldsymbol{d}}_j) \tag{93}$$

From the comparison between the terms of **Eq.**(92) and those of **Eq.**(93), the followings formulas are finally obtained.

$$\widehat{\boldsymbol{P}}_i=\boldsymbol{P}_i+\sum_{j\in C(i)}\boldsymbol{U}_j^i\widehat{\boldsymbol{P}}_j\boldsymbol{V}_i\boldsymbol{S}_i^j{\boldsymbol{V}_i}^{-1} \tag{94}$$

$$\widehat{\boldsymbol{d}}_i=\boldsymbol{d}_i+\sum_{j\in C(i)}\boldsymbol{U}_j^i(\widehat{\boldsymbol{P}}_j\boldsymbol{V}_i\boldsymbol{c}_j+\widehat{\boldsymbol{d}}_i) \tag{95}$$

D. Structure of 6×18 matrix $\hat{\mathbf{H}}_j$ in Eq.(71)

Let us writhe down 6×18 matrix $\widehat{\boldsymbol{H}}_j$ into three 6×6 block matrices $\widehat{\boldsymbol{K}}_j$, $\widehat{\boldsymbol{D}}_j$ and $\widehat{\boldsymbol{M}}_j$ as follows:

$$\widehat{\boldsymbol{H}}_j = \begin{bmatrix}\widehat{\boldsymbol{K}}_j & \widehat{\boldsymbol{D}}_j & \widehat{\boldsymbol{M}}_j\end{bmatrix} = \sum_{k\in\hat{\mathscr{C}}(j)}\boldsymbol{A}_k^{j\,-T}[\boldsymbol{O}_6 \quad \boldsymbol{D}_k \quad \boldsymbol{M}_k]{\boldsymbol{S}_k}^{-1}{\boldsymbol{X}_k^{p(k)}}^{-1}\boldsymbol{S}_{p(k)} \tag{96}$$

By substituting the each component of $\boldsymbol{X}$ and $\boldsymbol{S}$ according to **Eq.**(17) and **Eq.**(24), we can have:

$$\widehat{\boldsymbol{M}}_j = \boldsymbol{M}_j+\sum_{k\in C(j)}\boldsymbol{A}_k^{j\,-T}\widehat{\boldsymbol{M}}_k\boldsymbol{A}_k^{j\,-1} \tag{97}$$

$$\widehat{\boldsymbol{D}}_j = \boldsymbol{D}_j+\sum_{k\in C(j)}\boldsymbol{A}_k^{j\,-T}\left(\widehat{\boldsymbol{D}}_k-\widehat{\boldsymbol{M}}_k[\boldsymbol{v}_k^j\boldsymbol{.}\,]\right)\boldsymbol{A}_k^{j\,-1} \tag{98}$$

$$\widehat{\boldsymbol{K}}_j = \boldsymbol{O}_6 \tag{99}$$

References

[1] J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni. SOFA - an Open Source Framework for Medical Simulation. In *Proc. of Medicine Meets Virtual Reality*, volume 125, pages 13–18, 2007.

[2] K. Ayusawa, Y. Ikegami, and Y. Nakamura. Simultaneous global inverse kinematics and geometric parameter identification of human skeletal model from motion capture data. *Mechanism and Machine Theory*, 74:274–284, 2014.

[3] K. Ayusawa, G. Venture, and Y. Nakamura. Identifiability and identification of inertial parameters using the underactuated base-link dynamics for legged multibody systems. *Int. J. of Robotics Research*, 33(3):446–468, 2014.

[4] K. Ayusawa, M. Morisawa, and E. Yoshida. Motion retargeting for humanoid robots based on identification to preserve and reproduce human motion features. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2774–2779, 2015.

[5] K. Ayusawa, A. Rioux, G. Yoshida, E. Venture, and G. Maxime. Generating persistently exciting trajectory based on condition number optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 6518–6524, 2017.

[6] V. Bonnet, P. Fraisse, A. Crosnier, M. Gautier, A. González, and G. Venture. Optimal exciting dance for identifying inertial parameters of an anthropomorphic structure. *IEEE Trans. on Robotics*, 32(4):823–836, 2016.

[7] S. L. Delp and J. P. Loan. A computational framework for simulating and analyzing human and animal movement. *IEEE Computing in Science and Engineering*, 2(5):46–55, 2000.

[8] A. Escande, N. Mansard, and P.-B. Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *Int. J. of Robotic Research*, 33(7): 1006–1028, 2014.

[9] R. Fletcher. *Practical Methods of Optimization; (2Nd Ed.)*. Wiley-Interscience, New York, NY, USA, 1987.

[10] M. Gleicher. Retargetting motion to new characters. In *Proc. of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 33–42, 1998.

[11] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Resolved momentum control: humanoid motion planning based on the linear and angular momentum. In *Proc. of the 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1644–1650, 2003.

[12] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1620–1626, 2003.

[13] W. Khalil and E. Dombre. *Modeling, identification and control of robots*. Hermès Penton, London-U.K, 2002.

[14] O. Khatib. A unified approach for motion and force control of robotic manipulators: the operational space formulation. *IEEE J. Robotics and Automation*, 3(1): 43–53, 1987.

[15] A.G. Kirk, J.F. O'Brien, and D.A. Forsyth. Skeletal parameter estimation from optical motion capture data. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 782–788, 2005.

[16] J. Lim, I. Lee, I. Shim, H. Jung, H. Joe, H. Bae, O. Sim, J. Oh, T. Jung, S. Shin, K. Joo, M. Kim, K. Lee, Y. Bok, D.-G. Choi, B. Cho, S. Kim, J. Heo, I. Kim, J. Lee, I. Kwon, and J.-H. Oh. Robot system of drc-hubo+ and control strategy of team kaist in darpa robotics challenge finals. *J. of Field Robotics*, 2016.

[17] J. Luh, M. Walker, and R. Paul. Resolved-acceleration control of mechanical manipulators. *IEEE Trans. on Automatic Control*, 25(3):468–474, 1980.

[18] K. Miura, E. Yoshida, Y. Kobayashi, Y. Endo, F. Kanehioro, K. Homma, I. Kajitani, Y. Matsumoto, and T. Tanaka. Humanoid robot as an evaluator of assistive devices. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 671–677, 2013.

[19] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization*. Addison Wesley Publishing Company, 1991.

[20] Y. Nakamura, K. Yamane, K. Fujita, and I. Suzuki. Somatosensory computation for man-machine interface from motion-capture data and musculoskeletal human model. *IEEE Trans. on Robotics*, 21(1):58–66, 2005.

[21] F.-C. Park, J.-E. Bobrow, and S.-R. Ploen. A lie group formulation of robot dynamics. *I. J. of Robotics Research*, 14(6):609–618, 1995.

[22] N.S. Pollard, J.K. Hodgins, M.J. Riley, and C.G. Atkeson. Adapting human motion for the control of a humanoid robot. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1390–1397, 2002.

[23] G.-A. Sohl and J.-E. Bobrow. A recursive multibody dynamics and sensitivity algorithm for branched kinematic chains. *J. Dyn. Sys., Meas., Control*, 123(3):391–399, 2000.

[24] T. Sugihara and Y. Nakamura. Whole-body cooperative balancing of humanoid robot using COG jacobian. In *Proc. of the 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2575–2580, 2002.

[25] W. Suleiman, E. Yoshida, F. Kanehiro, J-P. Laumond, and A. Monin. Human motion imitation by humanoid robot. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2967–2704, 2008.

[26] K. Yamane and Y. Nakamura. Natural motion animation through constraining and deconstraining at will. *IEEE Trans. on Visualization and Computer Graphics*, 9(3): 352–360, 2003.