Proceedings 2005 IEEE International Symposium on
Computational Intelligence in Robotics and Automation
June 27-30, 2005, Espoo, Finland

The Kinematic Hessian and Higher Derivatives

Arjang Hourtash
Member IEEE
Simplify Robotics, Inc.
Houston, Texas, U.S.A.
arjang@simplifyrobotics.com

***Abstract*— A number of problems in manipulator analysis and control call for the second derivative of the joint- to work-space kinematic mapping of serial or branched manipulators. A derivation of all derivatives of the kinematic mapping is presented, including the second derivative namely the Hessian tensor. A fast formulation for its computation is derived which is based on components of the Jacobian matrix. The resulting formulae are verified symbolically with differentiation, and showcased numerically in Taylor series approximations and in a singularity escapability analysis for the example of the International Space Station's Canadarm2, a.k.a. the Space Station Remote Manipulator System (SSRMS).**

***Index Terms*— kinematic derivatives, singularity, self-motion, escapability, rate control**

I. Introduction

For manipulators, the kinematic mapping $f(q)$ relates a vector of joint variables to coordinate frames of interest, attached somewhere on the manipulator body. In most serial applications, the coordinate frame of interest is chosen to be the end-effector (EE) local frame. Since the joint variables directly influence this frame, and it is with this frame that the manipulator interacts with the world, the kinematic mapping $f(q)$ is critical to manipulation. Since the 1960s, resolved motion rate control in various forms has been used for guiding this frame of interest [1]. This approach linearizes the fundamentally nonlinear kinematic mapping $f(q)$ by using its first partial derivative, the Jacobian matrix. In the past several decades, many enhancements of this original approach have been devised. A number of applications require the use of the second derivative of $f(q)$, the Hessian H, although these are not as common. For brevity, the following assumes that the frame of interest is the EE pose. Consider the following applications of H:

(1) First, in resolved acceleration control, the objective is to calculate joint accelerations $\ddot{q}$ based on the desired motion profile of the EE pose. This relationship can be derived as follows:

$$\dot{f}(q) = J\dot{q} \qquad (1)$$

$$\ddot{f}(q) = J\ddot{q} + \dot{J}\dot{q} = J\ddot{q} + \dot{q}^T H\dot{q} \qquad (2)$$

where $J = \partial f(q)/\partial q$, $H = \partial^2 f(q)/\partial q^2$, and the elements of $(\dot{q}^T H)$ are organized so that they are conformal when multiplying with $\dot{q}$. These can be combined to express $\ddot{q}$ in terms of J, H, along with $f(q)$ and its derivatives, which can further be combined with a dynamics model to arrive at joint torques.

(2) A second application is the Quadratic Rate Control which is useful in the vicinity of manipulator singularities [8]. In this application, a complex joint rate (rather than a real one) is elegantly produced even when the desired EE rate is outside of the real range space of the Jacobian. Furthermore, since the linear Taylor series approximation breaks down at singularities, a quadratic approximation is used in the form

$$\Delta f(q) = J\Delta q + \frac{1}{2}\Delta q^T H \Delta q \qquad (3)$$

where the elements of $(\Delta q^T H)$ are organized so that they are conformal when multiplying with Δq. Next, a multivariate Newton-Raphson technique is used to solve for Δq given $\Delta f(q)$.

(3) A third application is one with limited computational resources, such as a real-time control system with many other subsystems to support. In such a case, it would be useful to avoid unnecessary Jacobian and post Jacobian calculations. One approach may be to choose a threshold based on some special norm $\|\Delta J_{t_o}\|$, which must be surpassed before Jacobian and post Jacobian calculations, along with t_o are refreshed. This query may simply involve a running sum of Δq_{t_i} in $\|H_{t_o}(\sum_{t_i} \Delta q_{t_i})\| \geq \|\Delta J_{t_o}\|$.

(4) In a fourth application, Bedrossian and Flueckiger [2]-[4], and later O'Neil, Seng, and Chen [5]-[7], used the Hessian to classify singularities as *escapable* or otherwise. Escapability is defined as whether or not self-motion reconfigures the manipulator to escape a singularity. This analysis shall be highlighted in the case studies at the end of this paper, and is therefore only briefly mentioned here.

Although the Hessian is used in these applications, a satisfactory and comprehensive derivation of the Hessian and higher partial derivatives of $f(q)$ is not widely available. This paper presents such a derivation for manipulators consisting of revolute and prismatic joints. The derived expressions are computationally thrifty since they are all based on the component calculations of the Jacobian. The derivations for the Hessian have been verified using symbolic differentiation. Additionally, a numerical case study is presented in which Taylor series approximations of $f(q + \Delta q)$, are compared with the exact solutions to illustrate the effect of the Hessian term. Finally, a numerical escapability analysis has been performed for the SSRMS.

0-7803-9355-4 / 05 / $20.00 ©2005 IEEE.

II. General Partial Derivatives

A. General Derivatives

The kinematic mapping

$$X = f(q) \tag{4}$$

of a serial link or a branched manipulator's joint- to work- spaces $\mathcal{R}^n \rightarrow \mathcal{R}^m$ is constructed from affine and trigonometric functions of the joint variables. In this work, orientations are considered as 3×3 rotation matrices for differentiability, although each of them only contributes 3 independent variables to the workspace of dimension m. Furthermore, the derivatives of rotations are expressed as 3×1 directions for angular velocities, accelerations, and so on. For brevity, the mathematics of rotations in various forms along with their singularities are not considered. Therefore, this kinematic mapping is considered to be continuous and differentiable, almost everywhere, for all higher derivatives.

In general, the n-th partial derivative forms an $(n+1)$-th order tensor. For example, the first partial derivative forms an $m \times n$ matrix, the second partial derivative forms an $m \times n \times n$ 3rd order tensor, and so on. Since the number of scalar elements for a Hessian can be large, computational efficiency is an important consideration.

This section develops formulations of general derivatives, whose component calculations can be reused for progressively higher derivatives, thereby reducing computational overhead as much as possible. These are used to first verify well known formulae for the Jacobian, and to construct new ones for the Hessian. The rest of this section discusses serial link manipulators, however these results can in general be extended to branched manipulators, by constructing a serial link formulation for each branch.

For the following, the kinematic mapping function in (4) is assumed to describe the position and orientation, collectively named the pose, of the EE with respect to the manipulator base. Note however, that any other frame of interest can be considered instead. The EE pose can be related to the manipulator joint variables using 4×4 homogeneous transformation matrices T, each of which is a composite of 3×1 positions p and 3×3 orientations R:

$$ {}^{i}_{j}T = \left(\begin{array}{c|c} {}^{i}_{j}R & {}^{i}p_j \\ \hline 0_{1\times 3} & 1 \end{array}\right) \tag{5}$$

In the following, sub- and super- scripts denote the following reference frames: b for the base, 0 for a fixed frame prior to the joint 1 frame, $1 \cdots n$ for local frames of the links actuated by joints $1 \cdots n$, and e for the EE. As an example, when the vector $[{}^{j}p^T \; 1]^T$ constructed from point ${}^{j}p$ in frame j is premultiplied by ${}^{i}_{j}T$, the result is $[{}^{i}p^T \; 1]^T$ expressing the point in frame i. Therefore, ${}^{i}_{j}T$ can be used to transform points from frames j to i. Given the above definitions, the EE pose becomes

$${}^{b}_{e}T = {}^{b}_{0}T \; {}^{0}_{1}T(q_1) \; \cdots \; {}^{n-1}_{n}T(q_n) \; {}^{n}_{e}T \tag{6}$$

where the notation ${}^{i-1}_{i}T(q_i)$ indicates that the coordinate transformation associated with the i-th joint is only dependent on its joint variable q_i. In this work, it is assumed that a joint is either revolute or prismatic, but not both. Also, if the generic joint variable q_i is associated with a revolute joint, it is referred to more specifically using θ_i, and if prismatic, then using ϕ_i.

Having decomposed the kinematic mapping function to its component Degrees of Freedom (DOF) in (6), the partial derivatives of any order with respect to (w.r.t.) any set of joint variables can be given as follows.

$$\frac{\partial^{(o_1)+\cdots+(o_n)} \, {}^{b}_{e}T}{\partial q_1^{(o_1)} \cdots \partial q_n^{(o_n)}} = {}^{b}_{0}T \, \frac{d^{(o_1)}T_1}{dq_1^{(o_1)}} \cdots \frac{d^{(o_n)}T_n}{dq_n^{(o_n)}} \, {}^{n}_{e}T \tag{7}$$

where $\partial^{(o_i)}/\partial q_i^{(o_i)}$ and $d^{(o_i)}/dq_i^{(o_i)}$ are the (o_i)-th partial and total derivatives w.r.t. q_i, respectively, and T_i is an abbreviation for ${}^{i-1}_{i}T(q_i)$. **In this work, the solutions to the general higher derivatives in (7) are presented generically by expressing the derivatives to (5) for revolute and prismatic joints, and leaving it to the reader to construct specific solutions.**

B. General Derivatives for Revolute Joints

For revolute joints, only the upper left of (5) is a variable. Therefore, the derivatives become:.

$$\frac{d^{(o_i)}({}^{i-1}_{i}T(\theta_i))}{d\theta_i^{(o_i)}} = \left(\begin{array}{c|c} d^{(o_i)}({}^{i-1}_{i}R(\theta_i))/d\theta_i^{(o_i)} & 0 \\ \hline 0 & 0 \end{array}\right) \tag{8}$$

Due to the rotary nature of revolute joints,

$$R(\theta_i) = rot({}^{i-1}\hat{a}_i, \theta_i) \tag{9}$$

where $rot(\hat{a}, b)$ is the operator of rotations about the $\hat{a}$ axis, by the amount b, and $({}^{i-1}\hat{a}_i)$ is the axis of rotation of the i-th joint frame in the $(i-1)$-th reference frame.

The derivative of a rotation matrix can be shown to be as follows:

$$d({}^{i}_{j}R)/d\theta = \lim_{\Delta\theta \to 0} \frac{{}^{i}_{j}R(\theta + \Delta\theta) - {}^{i}_{j}R(\theta)}{\Delta\theta} \tag{10}$$

Noting that

$${}^{i}_{j}R(\theta + \Delta\theta) = rot({}^{i}\hat{a}_j, \Delta\theta) \; {}^{i}_{j}R(\theta) \tag{11}$$

Furthermore, from Rodriguez' rotation formula,

$$rot({}^{i}\hat{a}_j, \Delta\theta) = I + \sin(\Delta\theta)({}^{i}\tilde{a}_j) + (1 - \cos(\Delta\theta))({}^{i}\tilde{a}_j)^2 \tag{12}$$

where $({}^{i}\tilde{a}_j)$ is the skew-symmetric cross-product matrix associated with unit vector $({}^{i}\hat{a}_j)$. Small angle approximations of $\Delta\theta$ lead to

$$rot({}^{i}\hat{a}_j, \Delta\theta) \approx I + (\Delta\theta)({}^{i}\tilde{a}_j) \tag{13}$$

So (10) becomes

$$\begin{aligned} d({}^{i}_{j}R)/d\theta &= \lim_{\Delta\theta \to 0} \frac{I + (\Delta\theta)({}^{i}\tilde{a}_j) - I}{\Delta\theta} \; {}^{i}_{j}R(\theta) \\ &= ({}^{i}\tilde{a}_j) \; {}^{i}_{j}R(\theta) \end{aligned} \tag{14}$$

Replacing i and j by $(i-1)$ and i, respectively, and repeating the above procedure (o_i) many times, produces a similar result with (o_i) power of $({}^{i-1}\tilde{a}_i)$:

$$\begin{aligned}\frac{d^{(o_i)}({}^{i-1}_{\ i}R(\theta_i))}{d\theta_i^{(o_i)}} &= ({}^{i-1}\tilde{a}_i)^{o_i}\ {}^{i-1}_{\ i}R(\theta_i) \\ &:= ({}^{i-1}\tilde{a}_i)^{(o_i-1)}({}^{i-1}\hat{a}_i) \end{aligned} \quad (15)$$

where the second result expresses the derivative of the rotation matrix in terms of the rotation axis vector.

An interesting interaction between rotation ${}^{i-1}_{\ i}R$ and cross-product $({}^{i}\tilde{a})$ matrices should be noted. For an arbitrary vector $({}^{i}{}_{v})$, the following holds:

$$\begin{aligned} & {}^{i-1}_{\ i}R\ ({}^{i}\tilde{a})\ ({}^{i}{}_{v}) \\ =\ & {}^{i-1}_{\ i}R(({}^{i}\hat{a}) \times ({}^{i}{}_{v})) \\ =\ & ({}^{i-1}_{\ i}R\ ({}^{i}\hat{a})) \times ({}^{i-1}_{\ i}R\ ({}^{i}{}_{v})) \\ =\ & ({}^{i-1}_{\ i}R\ ({}^{i}\hat{a})) \times ({}^{i-1}_{\ i}R\ ({}^{i}{}_{v}) + ({}^{i-1}p_i) - ({}^{i-1}p_i)) \\ =\ & ({}^{i-1}\hat{a}) \times ({}^{i-1}{}_{v} - {}^{i-1}p_i) \\ =\ & ({}^{i-1}\tilde{a})\ ({}^{i-1}{}_{v} - {}^{i-1}p_i) \end{aligned} \quad (16)$$

where the reader is directed to note the reference frames. When $({}^{i-1}{}_{v})$ is a column of a rotation matrix, $({}^{i-1}p_i) = 0$. Therefore, the following must also hold:

$$({}^{i-1}_{\ i}R)\ ({}^{i}\tilde{a})\ ({}^{\ i}_{i+1}R) = ({}^{i-1}\tilde{a})\ ({}^{i-1}_{i+1}R) \quad (17)$$

hence

$$\begin{aligned} & {}^{b}_{0}R\ \frac{d^{(o_1)}({}^{0}_{1}R(\theta_1))}{d\theta_1^{(o_1)}} \cdots \frac{d^{(o_i)}({}^{i-1}_{\ i}R(\theta_i))}{d\theta_i^{(o_i)}} \\ =\ & {}^{b}_{0}R\ (({}^{0}\tilde{a}_1)^{o_1}\ {}^{0}_{1}R(\theta_1)) \cdots (({}^{i-1}\tilde{a}_i)^{o_i}\ {}^{i-1}_{\ i}R(\theta_i)) \\ =\ & ({}^{b}\tilde{a}_1)^{o_1} \cdots ({}^{b}\tilde{a}_i)^{o_i}\ {}^{b}_{0}R^{0}_{1}R(\theta_1) \cdots {}^{i-1}_{\ i}R(\theta_i) \\ =\ & ({}^{b}\tilde{a}_1)^{o_1} \cdots ({}^{b}\tilde{a}_i)^{o_i}\ {}^{b}_{i}R(\theta_1, \cdots, \theta_i) \end{aligned} \quad (18)$$

The significance of (18) is that when (15) is substituted into (7), the skew symmetric matrices $\tilde{a}$ can be pulled to the left of all rotation matrices R for a cleaner result.

C. General Derivatives for Prismatic Joints

Next, the derivatives of (5) for prismatic joints are considered, in which case only the upper right of (5) is a variable. Therefore, the derivatives become:

$$\frac{d^{(o_i)}({}^{i-1}_{\ i}T(\phi_i))}{d\phi_i^{(o_i)}} = \left(\begin{array}{c|c} 0 & d^{(o_i)}({}^{i-1}p_i(\phi_i))/d\phi_i^{(o_i)} \\ \hline 0 & 0 \end{array}\right) \quad (19)$$

Due to the linear nature of prismatic joints,

$${}^{i-1}p_i = \phi_i({}^{i-1}\hat{a}_i) \quad (20)$$

and its derivatives are:

$$\frac{d^{(o_i)}({}^{i-1}p_i(\phi_i))}{d^{(o_i)}\phi_i} = \begin{cases} {}^{i-1}\hat{a}_i & o_i = 1 \\ 0_{3\times 1} & o_i \geq 2 \end{cases} \quad (21)$$

where $({}^{i-1}\hat{a}_i)$ is the prismatic axis of the i-th joint in the $(i-1)$-th reference frame. Note that higher than first derivatives of (5) for prismatic joints become $0_{4\times 4}$. In other words, a prismatic joint has a linear joint- to work- space mapping, i.e. a mapping without curvature.

D. Structure of General Derivatives

The structure of $d^{(o_i)}T_i/dq_i^{(o_i)}$ in (7) lends itself to certain simplifications. When joint i is revolute, i.e. $q_i = \theta_i$,

$$\frac{d^{(o_i)}({}^{i-1}_{\ i}T(\theta_i))}{d\theta_i^{(o_i)}} = \left(\begin{array}{c|c} - & 0 \\ \hline 0 & 0 \end{array}\right) \quad (22)$$

where $(-)$ indicates a non-zero partition. When other T matrices are pre- and post- multiplied by this structure, the effect is as follows:

$$T_a\left(\begin{array}{c|c} - & 0 \\ \hline 0 & 0 \end{array}\right)T_b = \left(\begin{array}{c|c} - & x \\ \hline 0 & 1 \end{array}\right)\left(\begin{array}{c|c} - & 0 \\ \hline 0 & 0 \end{array}\right)\left(\begin{array}{c|c} - & - \\ \hline 0 & 1 \end{array}\right) \quad (23)$$

where x indicates that the translation partition of the left matrix does not affect the result. Therefore in all transformations in (7) left of the rightmost revolute joint derivative, the translation partition is inconsequential and can be ignored from calculations.

Similarly, when joint i is prismatic, i.e. $q_i = \phi_i$,

$$\frac{d({}^{i-1}_{\ i}T(\phi_i))}{d\phi_i} = \left(\begin{array}{c|c} 0 & - \\ \hline 0 & 0 \end{array}\right) \quad (24)$$

When other T matrices are pre- and post- multiplied by this structure, the effect is as follows:

$$T_a\left(\begin{array}{c|c} 0 & - \\ \hline 0 & 0 \end{array}\right)T_b = \left(\begin{array}{c|c} - & x \\ \hline 0 & 1 \end{array}\right)\left(\begin{array}{c|c} 0 & - \\ \hline 0 & 0 \end{array}\right)\left(\begin{array}{c|c} x & x \\ \hline 0 & 1 \end{array}\right) \quad (25)$$

Therefore, all transformations in (7) right of the leftmost prismatic joint derivative, along with *all translations*, are inconsequential and can be ignored from calculations.

E. Jacobian for Revolute Joint Variables

Next, the Jacobian matrix is constructed as first the partial derivatives of (6). In this subsection, the differential variables are considered to be revolute joint variables. Although these Jacobian components are well known, their derivation serves as a good example for demonstrating the previous development. When considering partial derivatives with respect to θ_i, it is useful to rewrite (6) more briefly as:

$${}^{b}_{e}T = {}^{b}_{i}T(\theta_i)\ {}^{i}_{e}T \quad (26)$$

where the other joint variables affecting ${}^{b}_{i}T$ and ${}^{i}_{e}T$ are treated as constants for the purpose of the $\partial/\partial\theta_i$ operator. From (18),

$$\begin{aligned}\frac{\partial({}^{b}_{e}T)}{\partial\theta_i} &= \left(\begin{array}{c|c} ({}^{b}\tilde{a}_i)\ {}^{b}_{i}R(\theta_i) & 0 \\ \hline 0 & 0 \end{array}\right)\left(\begin{array}{c|c} {}^{i}_{e}R & {}^{i}p_e \\ \hline 0 & 1 \end{array}\right) \\ &= \left(\begin{array}{c|c} ({}^{b}\tilde{a}_i)\ {}^{b}_{e}R & ({}^{b}\tilde{a}_i)\ {}^{b}_{i}R(\theta_i){}^{i}p_e \\ \hline 0 & 0 \end{array}\right) \end{aligned} \quad (27)$$

For the translation component, a more convenient expression can be obtained by making the following substitutions:

$${}^{b}_{i}R(\theta_i){}^{i}p_e = ({}^{b}_{i}R(\theta_i){}^{i}p_e + {}^{b}p_i) - {}^{b}p_i = {}^{b}p_e - {}^{b}p_i \quad (28)$$

Expressing the rotation term using the rotation axis vector leads to the Jacobian column for revolute joints:

$$\frac{\partial X}{\partial\theta_i} = \begin{pmatrix} {}^{b}\hat{a}_i \times ({}^{b}p_e - {}^{b}p_i) \\ {}^{b}\hat{a}_i \end{pmatrix} \quad (29)$$

This is in agreement with the well known formulation for the Jacobian components for revolute joints.

F. Jacobian for Prismatic Joint Variables

Next, the components of the Jacobian matrix are constructed for when the differential variables are prismatic joint variables. When considering partial derivatives with respect to ϕ_i, it is useful to rewrite (6) more briefly as:

$$ {}^b_eT = {}^b_iT(\phi_i)\ {}^i_eT \tag{30}$$

From (21),

$$\begin{aligned} \frac{\partial({}^b_eT)}{\partial\phi_i} &= \left(\begin{array}{c|c} 0 & {}^b\hat{a}_i \\ \hline 0 & 0 \end{array}\right)\left(\begin{array}{c|c} {}^i_eR & {}^ip_e \\ \hline 0 & 1 \end{array}\right) \\ &= \left(\begin{array}{c|c} 0 & {}^b\hat{a}_i \\ \hline 0 & 0 \end{array}\right) \end{aligned}\tag{31}$$

which can be rewritten as:

$$\frac{\partial X}{\partial\phi_i} = \begin{pmatrix} {}^b\hat{a}_i \\ 0_{3\times1} \end{pmatrix} \tag{32}$$

Again, this is in agreement with the well known formulation for the Jacobian components for prismatic joints.

G. Hessian for Revolute-Revolute Joint Variables

Next the components of the Hessian tensor are constructed as the second partial derivatives of (6) when both differential variables are revolute joint variables. When considering partial derivatives with respect to θ_i and θ_j, it is useful to rewrite (6) more briefly as:

$$\begin{aligned} {}^b_eT &= {}^b_iT(\theta_i)\ {}^i_jT(\theta_j)\ {}^j_eT &&, i<j \\ {}^b_eT &= {}^b_iT(\theta_i)\ {}^i_eT &&, i=j \end{aligned}\tag{33}$$

From (18),

$$\frac{\partial^2({}^b_eT)}{\partial\theta_i\partial\theta_j} = \left(\begin{array}{c|c} \partial^2({}^b_eR)/\partial\theta_i\partial\theta_j & \partial^2({}^bp_e)/\partial\theta_i\partial\theta_j \\ \hline 0 & 0 \end{array}\right) \tag{34}$$

where $i \le j$, and

$$\frac{\partial^2({}^b_eR)}{\partial\theta_i\partial\theta_j} = ({}^b\tilde{a}_i)\ ({}^b\tilde{a}_j)\ {}^b_eR \tag{35}$$

$$\begin{aligned} \frac{\partial^2({}^bp_e)}{\partial\theta_i\partial\theta_j} &= ({}^b\tilde{a}_i)\ ({}^b\tilde{a}_j)\ {}^b_jR\ {}^jp_e \\ &= ({}^b\tilde{a}_i)\ ({}^b\tilde{a}_j)({}^bp_e - {}^bp_j) \end{aligned}\tag{36}$$

which includes the case $i = j$. Expressing the rotation using the rotation axis vector leads to:

$$\frac{\partial^2 X}{\partial\theta_i\partial\theta_j} = \begin{pmatrix} {}^b\hat{a}_i \times ({}^b\hat{a}_j \times ({}^bp_e - {}^bp_j)) \\ {}^b\hat{a}_i \times {}^b\hat{a}_j \end{pmatrix}, \ i \le j \tag{37}$$

Due to the symmetry of the Hessian, the values when $i > j$ are identical to those when $i < j$. Note that when $i = j$, the rotation component is ${}^b\hat{a}_i \times {}^b\hat{a}_i = 0_{3\times1}$.

H. Hessian for Prismatic-Prismatic Joints Variables

Next the Hessian components are considered when both differential variables are prismatic joint variables. When considering partial derivatives with respect to ϕ_i and ϕ_j, it is useful to rewrite (6) more briefly as:

$$\begin{aligned} {}^b_eT &= {}^b_iT(\phi_i)\ {}^i_jT(\phi_j)\ {}^j_eT &&, i<j \\ {}^b_eT &= {}^b_iT(\phi_i)\ {}^i_eT &&, i=j \end{aligned}\tag{38}$$

From (21), when $i = j$, $\partial^2({}^b_eT)/\partial\phi_i^2 = 0_{4\times4}$, and

$$\frac{\partial^2({}^b_eT)}{\partial\phi_i\partial\phi_j} = \left(\begin{array}{c|c} 0 & {}^b\hat{a}_i \\ \hline 0 & 0 \end{array}\right)\left(\begin{array}{c|c} 0 & {}^b\hat{a}_j \\ \hline 0 & 0 \end{array}\right)\ {}^j_eT = 0_{4\times4} \tag{39}$$

when $i \ne j$, resulting in the general expression:

$$\frac{\partial^2 X}{\partial\phi_i\partial\phi_j} = 0_{6\times1} \tag{40}$$

This indicates that the joint- to work- space mapping for prismatic joints is linear, and therefore has no curvature.

I. Hessian for Revolute-Prismatic Joint Variables

Next the components of the Hessian are constructed for when the differential variables involve both revolute and prismatic joint variables. When considering partial derivatives with respect to the revolute joint variable θ_i and the prismatic joint variable ϕ_j, it is useful to rewrite (6) more briefly as:

$$ {}^b_eT = {}^b_iT(\theta_i)\ {}^i_jT(\phi_j)\ {}^j_eT\ ,\ i<j \tag{41}$$

Note that from Subsection II-D, $\partial^2({}^b_eT)/\partial\theta_i\partial\phi_j = 0_{4\times4}$ when $i > j$, i.e. when the prismatic joint precedes the revolute; and due to the mixed joint types, the $i = j$ case is not meaningful. Combining (18) and (21) leads to:

$$\frac{\partial^2({}^b_eT)}{\partial\theta_i\partial\phi_j} = \left(\begin{array}{c|c} \partial^2({}^b_eR)/\partial\theta_i\partial\phi_j & \partial^2({}^bp_e)/\partial\theta_i\partial\phi_j \\ \hline 0 & 0 \end{array}\right) \tag{42}$$

where

$$\begin{aligned} \frac{\partial^2({}^b_eR)}{\partial\theta_i\partial\phi_j} &= 0_{3\times3} \\ \frac{\partial^2({}^bp_e)}{\partial\theta_i\partial\phi_j} &= ({}^b\tilde{a}_i)\ {}^b_iR\ ({}^i\hat{a}_j) = ({}^b\tilde{a}_i)({}^b\hat{a}_j),\ i<j \end{aligned}\tag{43}$$

This can be rewritten as:

$$\begin{aligned} \frac{\partial^2 X}{\partial\theta_i\partial\phi_j} &= \begin{pmatrix} {}^b\hat{a}_i \times {}^b\hat{a}_j \\ 0_{3\times1} \end{pmatrix} &&, i<j \\ \frac{\partial^2 X}{\partial\theta_i\partial\phi_j} &= 0_{6\times1} &&, i>j \end{aligned}\tag{44}$$

where i and j are the indices of the revolute and prismatic joints, respectively. Note that the $i < j$ and $i > j$ cases of (44) do not violate the symmetry of the Hessian since these two cases are associated with kinematically different manipulators (the revolute preceding the prismatic joint, or vice versa), rather than symmetric Hessian components of the same manipulator.

This section presented the foundation for obtaining the arbitrarily high order derivatives for the kinematic mapping in (4) for a manipulator with revolute and/or prismatic joints. This is detailed in (7), (8), (15), (18), (19), and (21). The above was demonstrated in deriving well known Jacobian expressions in (29) and (32), and was used for deriving Hessian expressions in (37), (40), and (44).

Additionally, all computational components of all arbitrarily high order derivatives of (4) can be expressed in terms of the joint axes $({}^b\hat{a}_i)$ and joint positions $({}^bp_i)$, all in the base frame. These calculations are necessary for the Jacobian, but can be reused for higher derivatives. Furthermore, it is worthwhile to note that for an n-DOF manipulator with $f(q)$ defined in $\mathcal{R}^m$, each $n \times n$ slice of the 3rd order Hessian tensor is a matrix of second derivatives, and is therefore symmetrical. In fact, symmetry is present in all even order partial derivative tensors, which can be useful in speeding up calculations. Also Subsection II-D describes how the structure of the matrices can be used for pruning unnecessary calculations. Finally, these results can be extended to branched manipulators, by reconstructing serial links for each branch.

III. Case Studies

A. Symbolic Verification

The above formulations were verified symbolically for a number of example manipulators using the GiNaC C++ library for computer algebra. Symbolic derivatives of the well known formulae for the Jacobian were calculated for manipulators with up to 5 DOF, and various sequences of revolute and prismatic joints, to obtain Hessian expressions. In all cases, $\partial^2 X/\partial q_i \partial q_j$ generated by the GiNaC software for both joint types, matched the Hessian expressions presented in this work.

As an example, consider a spatial stick-figure manipulator with 3 links of unit length. Each link's length axis is its local X-axis. The first two joints are revolute, rotating about the local Z- and Y- axes, respectively. The last joint is prismatic, extending along the X-axis. For this manipulator, the Jacobian J for the EE pose is

$$J = \begin{pmatrix} -(\phi_3+2)s_1c_2 - s_1 & -(\phi_3+2)c_1s_2 & c_1c_2 \\ (\phi_3+2)c_1c_2 + c_1 & -(\phi_3+2)s_1s_2 & s_1c_2 \\ 0 & -(\phi_3+2)c_2 & -s_2 \\ 0 & -s_1 & 0 \\ 0 & c_1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{45}$$

where the first and second 3 rows are partial derivatives of EE positions and orientations, respectively. Furthermore, $c_i = cos(\theta_i)$ and $s_i = sin(\theta_i)$. The Hessian H for this manipulator is expressed below, with H_i being the 3×3 slices corresponding to the derivatives of the rows of J:

$$H_1 = \begin{pmatrix} -(\phi_3+2)c_1c_2 - c_1 & (\phi_3+2)s_1s_2 & -s_1c_2 \\ (\phi_3+2)s_1s_2 & -(\phi_3+2)c_1c_2 & -c_1s_2 \\ -s_1c_2 & -c_1s_2 & 0 \end{pmatrix}$$

$$H_2 = \begin{pmatrix} -(\phi_3+2)s_1c_2 - s_1 & -(\phi_3+2)c_1s_2 & c_1c_2 \\ -(\phi_3+2)c_1s_2 & -(\phi_3+2)s_1c_2 & -s_1s_2 \\ c_1c_2 & -s_1s_2 & 0 \end{pmatrix}$$

$$H_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & (\phi_3+2)s_2 & -c_2 \\ 0 & -c_2 & 0 \end{pmatrix}$$

$$H_4 = 0_{3\times3}, \text{ except } H_4(1,2) = H_4(2,1) = -c_1$$

$$H_5 = 0_{3\times3}, \text{ except } H_5(1,2) = H_5(2,1) = -s_1$$

$$H_6 = 0_{3\times3} \tag{46}$$

For this and other examples, the Hessian formulae in this work matched the symbolic derivatives of the Jacobian.

B. Numerical Taylor Series Approximations

The previous derivations have been applied to a number of manipulators, including a simulated one with all prismatic joints, one with both revolute and prismatic joints, and the SSRMS. In these cases, the Hessian $H(q)$ was used to produce a more accurate Taylor series expansion, than the one with just the Jacobian $J(q)$:

$$f_1(q,\Delta q) = f(q) + J(q)\Delta q \tag{47}$$

$$f_2(q,\Delta q) = f_1(q,\Delta q) + \frac{\Delta q^T H(q) \Delta q}{2} \tag{48}$$

$$\Delta f_1(q,\Delta q) = f_1(q,\Delta q) - f(q+\Delta q) \tag{49}$$

$$\Delta f_2(q,\Delta q) = f_2(q,\Delta q) - f(q+\Delta q) \tag{50}$$

where in the above notation, the rotation components of f, f_1, and f_2 are handled appropriately using rotation matrix multiplications. Table I presents tabulated data from Δf_1 and Δf_2. These are presented as $|\Delta pos|$, the 2-norm of the EE position error, and |rot. angle|, the scalar magnitude of the EE orientation error. Note that $(\Delta f_1 - \Delta f_2)$ is positive whenever using the Hessian reduces the error.

The SSRMS, referred to as *ss* in the table, is a symmetrical slender space crane which has 7 revolute DOF. When fully extended, it is 17.014 meters from EE to base. The 2 configurations presented in Table I are: $\bar{q}_1 = 0^\circ_{7\times1}$, and $\bar{q}_2 = 45^\circ_{7\times1}$. Note that $\bar{q}_1$ is at an SSRMS singularity.

The *pm* manipulator has 7 prismatic DOF in 3D space. Since its Hessian is $H = 0_{6\times7\times7}$, it offers no improvement, i.e. $(\Delta f_1 - \Delta f_2)$, as is indicated in Table I. Therefore, its description is inconsequential and is omitted for brevity.

The *mx* manipulator has a mix of 7 prismatic (p) and revolute (r) DOF, in the following order: $\{p, r, p, r, p, r, p\}$. The joint axes are aligned with the following local coordinate axes: $\{X, Z, Y, Y, Y, Z, X\}$, and all links are 1 meter long, i.e. the manipulator is 8 meters long in its 0 configuration. The examined configurations are $\bar{q}_1 = [1m, 0^\circ, 1m, 0^\circ, 1m, 0^\circ, 1m]^T$, and $\bar{q}_2 = [1m, 45^\circ, 1m, 45^\circ, 1m, 45^\circ, 1m]^T$.

In Table I, k is the ratio of Δq to the range of joint motion, and is expressed as either 1% or 0.1%. Some observations validated expectations: First, Δf_2 improved as Δq becomes smaller. In the presented cases, shrinking Δq 10 times, reduced the error by 2-3 orders of magnitude when using the quadratic model. Second, comparing Δf_2 with $(\Delta f_1 - \Delta f_2)$ for the $|\Delta pos|$ shows that the difference between the exact and quadratic models is smaller than the difference between the quadratic and linear models.

Furthermore, this difference seems to grow in relative orders of magnitude as Δq becomes smaller, hinting to the fact that the quadratic model is converging faster toward the exact, than the linear model. A third but perhaps unexpected result is that incorporating the Hessian actually deteriorates the rotation error by a small amount. This may be caused by a possible high significance of the cubic and higher terms for rotations.

TABLE I
TAYLOR SERIES APPROXIMATIONS

robot	k	$\|\Delta$ pos$\|$ (meters)		$\|$ rot. angle $\|$ (rad)	
config	(%)	$\Delta f_1 - \Delta f_2$	Δf_2	$\Delta f_1 - \Delta f_2$	Δf_2
ss $\bar{q}_1$	1	3.51e-1	3.89e-2	0	2.21e-3
ss $\bar{q}_1$	0.1	3.93e-3	3.92e-5	0	2.22e-6
ss $\bar{q}_2$	1	2.15e-1	2.32e-2	-7.42e-4	3.28e-2
ss $\bar{q}_2$	0.1	2.37e-3	2.40e-5	-1.05e-7	3.23e-4
pm $\bar{q}_1$	1	0	5.66e-2	0	0
mx $\bar{q}_1$	1	5.39e-1	2.73e-2	-2.59e-5	4.43e-3
mx $\bar{q}_1$	0.1	5.64e-3	2.73e-5	-2.60e-9	4.41e-5
mx $\bar{q}_2$	1	5.27e-1	2.64e-2	-2.45e-5	4.44e-3
mx $\bar{q}_2$	0.1	5.51e-3	2.64e-5	-1.13e-9	4.42e-5

C. Singularity Escapability

From the introduction, escapability is defined as whether or not self-motion reconfigures the manipulator to escape a singularity. The SVD of the Jacobian $J = U\Sigma V^T$ is useful in isolating the unreachable directions in the workspace U_s, where $U = [U_r|U_s]$, as well as the null-space basis in the joint space V_s, where $V = [V_r|V_s]$. From [2], [5], the Taylor series can be used to express Δx as:

$$x(q+\Delta q) - x(q) = J(q)\Delta q + \frac{1}{2}\Delta q^T H(q)\Delta q + \cdots \quad (51)$$

During self-motion, the LHS of (51) vanishes, and $\Delta q = V_s b$, if $\Delta q \to 0$, where b is a vector of coefficients. In a singularity, the unreachable workspace directions are captured as ${U_s}^T\Delta x = {U_s}^T J\Delta q = 0$. Therefore, premultiplying (51) by ${U_s}^T$, produces a number of terms on the RHS, each of which must vanish. The first of these is the quadratic term

$$0 = b^T A b \quad (52)$$

where $A = {V_s}^T[{U_{s,i}}^T H(q)]V_s$, and $U_{s,i}$ is any vector from U_s. Escapability from a codimension 1 singularity of a redundant arm such as the SSRMS requires (52) to be satisfied with $b \neq 0$ for all $U_{s,i}$. Therefore, if A is definite for any $U_{s,i}$, then the singularity is inescapable [5].

Table II tabulates escapability results for some SSRMS singularities: The first two rows are for fully extended elbows. Note that not all extended elbow configurations are singularities. For the first row, escapability depends on the $[q_2, q_3]$ pair, whereas for the second, all combinations are escapable. A detailed analysis of the SSRMS singularities is beyond the scope of this paper, and is therefore omitted. However, it is noteworthy that the availability of the Hessian has expanded our analysis capability.

TABLE II
SINGULARITY ESCAPABILITY FOR THE SSRMS

$q_i, i \in \{1, \cdots, 7\}$ (degrees)							escape?
q_1	q_2	q_3	0	q_3	$-q_2$	q_7	No/Yes
q_1	q_2	q_3	0	$-q_3$	$-q_2$	q_7	Yes
q_1	± 90	q_3	q_4	q_5	± 90	q_7	Yes
q_1	q_2	-90	180	90	q_6	q_7	Yes
q_1	q_2	0	191.2866	0	q_6	q_7	No

IV. CONCLUSION

This paper presented derivations for arbitrarily high order partial derivatives of the kinematic mapping $f(q)$ for manipulators with revolute and prismatic joints. These general derivations were applied to reproduce well known formulae for the first partial derivatives, namely the Jacobian, and to produce new results for the second, namely the Hessian. The main results are summarized at the end of Section II. The expressions for the Hessian were verified by symbolically differentiating the Jacobian. The linear and quadratic approximations of the Taylor series were compared with the exact $f(q + \Delta q)$ to illustrate the impact of incorporating the the Hessian term. Furthermore, the Hessian was used for an escapability analysis for the SSRMS. For computational efficiency, (1) the higher derivative calculations reuse the Jacobian calculations, (2) the symmetry of the Hessian and even order derivatives further reduce the calculations, and (3) the structure of the matrices lend to pruning of unnecessary calculations. This work is applicable to branched manipulators, and serves as a reference for algorithms and analyses which involve second or higher order partial derivatives of $f(q)$.

REFERENCES

[1] D. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses", IEEE Transactions on Man-Machine Systems, Vol. MMS-10, No. 2, June 1969.

[2] N. Bedrossian, "Classification of Singular Configurations For Redundant Manipulators", Proc. IEEE Int'l Conf. Robotics and Automation, 1990, pp. 818-823.

[3] N. Bedrossian, K. Flueckiger, "Characterizing Spatial Redundant Manipulator Singularities", Proc. IEEE Int'l Conf. Robotics and Automation, 1991, pp. 714-719.

[4] K. Flueckiger, N. Bedrossian, "Unit Invariant Characterization of Spatial Redundant Manipulator Singularities", Proc. IEEE Int'l Conf. Robotics and Automation, 1992, pp. 409-414.

[5] J. Seng, K. O'Neil, Y.C. Chen, "Escapability of Singular Configuration for Redundant Manipulators via Self-Motion" , Proc. IEEE Int'l Conf. Intelligent Robotics and Systems, 1995, pp. 78-83.

[6] K. O'Neil, Y.C. Chen, J. Seng, "Desingularization of Resolved Motion Rate Control of Mechanisms", Proc. IEEE Int'l Conf. Robotics and Automation, 1996, pp. 3147-3154.

[7] K. O'Neil, Y.C. Chen, J. Seng, "Removing Singularities of Resolved Motion Rate Control of Mechanisms, Including Self-Motion", IEEE Trans. Robotics and Automation, Oct. 1997, Vol. 13, No. 5, pp. 741-751.

[8] E. Pohl, H. Lipkin, "A New Method of Robotic Rate Control Near Singularities", Proc. IEEE Int'l Conf. Robotics and Automation, 1991, pp. 1708-1713.