N90-29

Kinematic Sensitivity of Robot Manipulators

Marko I. Vuskovic
Department of Mathematical Sciences
San Diego State University
San Diego, CA 92182-0314

Abstract

Kinematic sensitivity vectors and matrices for open-loop, n degrees-of-freedom manipulators are derived. First-order sensitivity vectors are defined as partial derivatives of the manipulator's position and orientation with respect to its geometrical parameters. Considered is the four-parameter kinematic model, as well as the five-parameter model in case of nominally parallel joint axes. Sensitivity vectors are expressed in terms of coordinate axes of manipulator frames. Second-order sensitivity vectors, the partial derivatives of first-order sensitivity vectors, are also considered. It is shown that second-order sensitivity vectors can be expressed as vector products of the first-order sensitivity vectors.

1. Introduction

Sensitivity Theory plays an important role in Systems Theory and in Control Engineering. Its major part, Sensitivity Analysis, is studying the effects of small variations of system parameters on its dynamic behavior and performance criteria. This information can be used for identification of the system's mathematical model, and for the optimal design of the system's controller. Sensitivity theory is also concerned with methods of efficient generation of sensitivity functions in real-time, which can be used for adaptive control. The main results of Sensitivity theory with application in control were obtained in the early sixties. An excellent survey of the Sensitivity Theory at that time is given by two of its important contributors, Kokotovic and Rutman [9]. More recent overviews of the Sensitivity Theory are given by Cruz [3] and Frank [5].

Bearing in mind the substantial influence of the Sensitivity Theory on the development of Control Theory, the question can naturally be raised, whether it can play a similar role in Robotics. In fact, there are many areas in Robotics where sensitivity functions are implicitly used. An example is Robot Calibration, which has been established as an important discipline of Robotics [12], and which can be considered as a counterpart of System Identification, a discipline of System Theory. Finally, one of the most important quantities in Robotics, the manipulator Jacobian matrix, is a sensitivity matrix of the robot position and orientation with respect to joint angles.

The terms "sensitivity", or "kinematic sensitivity" is explicitly used in Robotics by Togai [13]. He has proposed the kinematic sensitivity matrix as a new quantitative measure for the capability for accurate positioning and orienting of a manipulator. This measure is proposed as an attribute complementing Yoshikawa's robot manipulability [15]. Asada and Hara [1] have also defined and analyzed the sensitivity of the actuator torque of the direct-drive arm with respect to the inertial loads. Both papers are dealing with sensitivity only partially, in the context of other problems, and without particular attention given to the problem of computing sensitivity functions in the general case.

This paper considers sensitivity vectors and matrices more generally, although restricted to Robot Kinematics. Sensitivity vectors are defined as partial derivatives of the manipulator's position and orientation with respect to its geometric parameters: link twists, link distances, link offsets and joint angles. Their explicit expressions are derived in terms of link coordinate axes. The case of nominally parallel joint axes, i.e. the five parameter model proposed by Hayati [6-8], is also considered. The sensitivities with respect to link twists about y-axes are derived for this case. Sensitivity vectors, or more precisely, the first-order sensitivity vectors, are then used to derive second-order sensitivity vectors, which are second-order partial derivatives of the manipulator position and orientation with respect to link parameters. It is shown that second-order sensitivity vectors can be entirely expressed in terms of first-order sensitivity vectors. The appendix supplied at the end of the paper reviews the basic formulas from robot kinematics which are used in the derivation of kinematic sensitivities. It also presents an efficient recursive algorithm for computation of link coordinate axes, for both the four and the five parameter models of forward kinematics.

2. Sensitivity vectors and matrices

Position and orientation of an n-DOF manipulator are characterized by its position vector $\mathbf{p} = [\,p_1\, p_2\, p_3\,]^T$ and orientation matrix $\mathbf{R} = [r_{ij}]_3$. These quantities can be referred to any link with respect to any coordinate system. We will consider position and orientation of the n-th link, called the wrist, with the 0-th link, called the base, as a reference coordinate system, i.e. $\mathbf{p} = {}^o\mathbf{p}_n$ and $\mathbf{R} = {}^o_n\mathbf{R}$.

The manipulator geometry is defined by its link parameters. We will use the modified four-parameter Denavit-Hartenberg model as proposed by Craig[2], in which the link parameters are: a_i (link distances), α_i (link twists), d_i (link offsets) and θ_i (joint angles). Therefore vector $\mathbf{p}$ and matrix $\mathbf{R}$ are functions of these parameters:

$$\mathbf{p} = \mathbf{p}(\mathbf{a},\alpha,\mathbf{d},\theta), \qquad \mathbf{R} = \mathbf{R}(\mathbf{a},\alpha,\mathbf{d},\theta) \tag{1}$$

where: $\mathbf{a} = [a_0\ a_1\ \dots\ a_{n-1}\ ]^T$, $\alpha = [\alpha_0\ \alpha_1\ \dots\ \alpha_{n-1}\ ]^T$, $\mathbf{d} = [d_1\ d_2\ \dots\ d_n\ ]^T$ and $\theta = [\theta_1\ \theta_2\ \dots\ \theta_n]^T$.

In order to study variations of $\mathbf{p}$ and $\mathbf{R}$ caused by small variations of link parameters we have to consider their partial derivatives $\partial\mathbf{p}/\partial c$ and $\partial\mathbf{R}/\partial c$, where c stands for any of the link parameters. The first partial derivative, $\partial\mathbf{p}/\partial c$, we call the *positional sensitivity vector*.

The orientation sensitivity is not so straightforward. The derivative $\partial\mathbf{R}/\partial c$ does not give convenient information about the variation of the manipulator's orientation. It would be more appropriate to express it in terms of three angles instead of a nine-element orientation matrix. Therefore we represent a small change of the manipulator's orientation through three infinitesimal orthogonal rotations $\Delta\varphi = [\Delta\varphi_1\ \Delta\varphi_2\ \Delta\varphi_3]^T$ about the axes of the base coordinate system. This can be written:

$$\mathbf{R}(c+\Delta c) = \Phi(\Delta\varphi)\,\mathbf{R}(c), \tag{2}$$

where

$$\Phi(\Delta\varphi) = \mathbf{rot}(e_3,\Delta\varphi_3)\ \mathbf{rot}(e_2,\Delta\varphi_2)\ \mathbf{rot}(e_1,\Delta\varphi_1). \tag{3}$$

The definition of the $\mathbf{rot}$ operator is given in the Appendix (see (A-4)). For sufficiently small $\Delta\varphi_i$, (3) becomes [11]:

$$\Phi(\Delta\varphi) \approx \mathbf{I} + \Lambda(\Delta\varphi) \tag{4}$$

where $\mathbf{I}$ is the 3×3 identity matrix and $\Lambda(\Delta\varphi)$ is a skew-symmetric operator (see (A-5)). Thus

$$\frac{\partial\mathbf{R}}{\partial c} = \lim_{\Delta c\to 0}\frac{\mathbf{R}(c+\Delta c) - \mathbf{R}(c)}{\Delta c} = \lim_{\Delta c\to 0}\frac{\Lambda(\Delta\varphi)}{\Delta c}\mathbf{R}(c) = \Lambda\!\left(\frac{\partial\varphi}{\partial c}\right)\mathbf{R}(c), \tag{5}$$

where

$$\frac{\partial\varphi}{\partial c} = \lim_{\Delta c\to 0}\frac{\Delta\varphi}{\Delta c}$$

we denote as the *orientation sensitivity vector*. The relation between the orientation sensitivity vector and the partial derivative of the orientation matrix is given by:

$$\Lambda\!\left(\frac{\partial\varphi}{\partial c}\right) = \frac{\partial\mathbf{R}}{\partial c}\mathbf{R}^{\mathrm{T}}. \tag{6}$$

Sensitivity vectors for various link parameters can be joined together to form the sensitivity matrices:

$$\mathbf{S}^p_a = \begin{bmatrix}\frac{\partial\mathbf{p}}{\partial a_0} & \cdots & \frac{\partial\mathbf{p}}{\partial a_{n-1}}\end{bmatrix},\quad \mathbf{S}^\varphi_a = \begin{bmatrix}\frac{\partial\varphi}{\partial a_0} & \cdots & \frac{\partial\varphi}{\partial a_{n-1}}\end{bmatrix},\quad \mathbf{S}^p_\alpha = \begin{bmatrix}\frac{\partial\mathbf{p}}{\partial\alpha_0} & \cdots & \frac{\partial\mathbf{p}}{\partial\alpha_{n-1}}\end{bmatrix},\quad \mathbf{S}^\varphi_\alpha = \begin{bmatrix}\frac{\partial\varphi}{\partial\alpha_0} & \cdots & \frac{\partial\varphi}{\partial\alpha_{n-1}}\end{bmatrix}$$

$$\mathbf{S}^p_d = \begin{bmatrix}\frac{\partial\mathbf{p}}{\partial d_1} & \cdots & \frac{\partial\mathbf{p}}{\partial d_n}\end{bmatrix},\quad \mathbf{S}^\varphi_d = \begin{bmatrix}\frac{\partial\varphi}{\partial d_1} & \cdots & \frac{\partial\varphi}{\partial d_n}\end{bmatrix},\quad \mathbf{S}^p_\theta = \begin{bmatrix}\frac{\partial\mathbf{p}}{\partial\theta_1} & \cdots & \frac{\partial\mathbf{p}}{\partial\theta_n}\end{bmatrix},\quad \mathbf{S}^\varphi_\theta = \begin{bmatrix}\frac{\partial\varphi}{\partial\theta_1} & \cdots & \frac{\partial\varphi}{\partial\theta_n}\end{bmatrix}. \tag{7}$$

3. Derivation of sensitivity vectors

Positional sensitivity vectors with respect to parameters a_j and d_j can be directly obtained if we express the manipulator position $\mathbf{p}$ explicitly in terms of these parameters. Such an expression is given in the Appendix. Since coordinate axes are independent of these parameters, (A-8) is giving:

$$\frac{\partial \mathbf{p}}{\partial a_j} = \mathbf{x}_j, \qquad \frac{\partial \mathbf{p}}{\partial d_j} = \mathbf{z}_j. \tag{8}$$

Hence

$$\boxed{\mathbf{S}^p_a = [\mathbf{x}_0\ \mathbf{x}_1 \ldots \mathbf{x}_{n-1}] \qquad \mathbf{S}^p_d = [\mathbf{z}_1\ \mathbf{z}_2 \ldots \mathbf{z}_n]} \tag{9}$$

In order to derive positional sensitivity vectors with respect to α_j and θ_j, we first find partial derivatives of coordinate axes with respect to these parameters. Combining (A-3) and (A-7) we can write:

$$\frac{\partial}{\partial \alpha_{i-1}} {}^{i-1}_{\ \ i}\mathbf{R} = \Lambda(\mathbf{e}_1)\ {}^{i-1}_{\ \ i}\mathbf{R}, \qquad \frac{\partial}{\partial \theta_i} {}^{i-1}_{\ \ i}\mathbf{R} = {}^{i-1}_{\ \ i}\mathbf{R}\ \Lambda(\mathbf{e}_3). \tag{10}$$

Now applying (10) and (A-6) to (A-8) and assuming $j<i$, we obtain:

$$\frac{\partial \mathbf{x}_i}{\partial \alpha_j} = \frac{\partial}{\partial \alpha_j}\left({}^{0}_{j}\mathbf{R}\ {}^{j}_{j+1}\mathbf{R}\ {}^{j+1}_{\ \ i}\mathbf{R}\ \mathbf{e}_1\right) = {}^{0}_{j}\mathbf{R}\ \Lambda(\mathbf{e}_1)\ {}^{j}_{i}\mathbf{R}\ \mathbf{e}_1 = \Lambda({}^{0}_{j}\mathbf{R}\mathbf{e}_1){}^{0}_{i}\mathbf{R}\ \mathbf{e}_1 = \Lambda(\mathbf{x}_j)\,\mathbf{x}_i = \mathbf{x}_j \times \mathbf{x}_i.$$

Thus:

$$\frac{\partial \mathbf{x}_i}{\partial \alpha_j} = \begin{cases} -\mathbf{x}_i \times \mathbf{x}_j & j<i \\ 0 & j \geq i \end{cases}. \tag{11}$$

In a similar way we can obtain other partial derivatives:

$$\frac{\partial \mathbf{z}_i}{\partial \alpha_j} = \begin{cases} -\mathbf{z}_i \times \mathbf{x}_j & j<i \\ 0 & j \geq i \end{cases}, \qquad \frac{\partial \mathbf{x}_i}{\partial \theta_j} = \begin{cases} -\mathbf{x}_i \times \mathbf{z}_j & j \leq i \\ 0 & j > i \end{cases}, \qquad \frac{\partial \mathbf{z}_i}{\partial \theta_j} = \begin{cases} -\mathbf{z}_i \times \mathbf{z}_j & j<i \\ 0 & j \leq i \end{cases}. \tag{12}$$

Applying now (11) and (12) to (A-8) we get:

$$\frac{\partial \mathbf{p}}{\partial \alpha_j} = \sum_{i=1}^{n} \frac{\partial \mathbf{x}_{i-1}}{\partial \alpha_j} a_{i-1} + \frac{\partial \mathbf{z}_i}{\partial \alpha_j} d_i = -\sum_{i=j+1}^{n} (\mathbf{x}_{i-1} a_{i-1} + \mathbf{z}_i d_i) \times \mathbf{x}_j. \tag{13}$$

Finally, substituting (A-9) into (13) gives us:

$$\frac{\partial \mathbf{p}}{\partial \alpha_j} = \mathbf{x}_j \times \mathbf{r}_j. \tag{14}$$

Similarly we get:

$$\frac{\partial \mathbf{p}}{\partial \theta_j} = \mathbf{z}_j \times \mathbf{r}_j, \tag{15}$$

thus:

$$\boxed{\mathbf{S}^p_\alpha = [\mathbf{x}_0 \times \mathbf{r}_0\ \ \mathbf{x}_1 \times \mathbf{r}_1 \ldots \mathbf{x}_{n-1} \times \mathbf{r}_{n-1}] \qquad \mathbf{S}^p_\theta = [\mathbf{z}_1 \times \mathbf{r}_1\ \mathbf{z}_2 \times \mathbf{r}_2 \ldots \mathbf{z}_{n-1} \times \mathbf{r}_{n-1}\ \ 0]} \tag{16}$$

In order to get orientation sensitivity vectors we first differentiate $\mathbf{R}$ with respect to α_j and θ_j:

$$\frac{\partial \mathbf{R}}{\partial \alpha_j} = \frac{\partial}{\partial \alpha_j}\left({}^{o}_{j}\mathbf{R}\ {}^{j}_{j+1}\mathbf{R}\ {}^{j+1}_{n}\mathbf{R}\right) = {}^{o}_{j}\mathbf{R}\,\Lambda(\mathbf{e}_1)\ {}^{j}_{j+1}\mathbf{R}\ {}^{j+1}_{n}\mathbf{R} = \Lambda({}^{o}_{j}\mathbf{R}\,\mathbf{e}_1)\,{}^{o}_{n}\mathbf{R} = \Lambda(\mathbf{x}_j)\,\mathbf{R}. \tag{17}$$

Substituting (17) into (6) gives us:

$$\Lambda\left(\frac{\partial \varphi}{\partial \alpha_j}\right) = \Lambda(\mathbf{x}_j),$$

which yields:

$$\frac{\partial \varphi}{\partial \alpha_j} = \mathbf{x}_j.$$

In a similar way we obtain:

$$\frac{\partial \varphi}{\partial \theta_j} = \mathbf{z}_j.$$

Since $\mathbf{R}$ does not depend on a_j and d_j it follows that:

$$\frac{\partial \varphi}{\partial a_j} = \frac{\partial \varphi}{\partial d_j} = 0. \tag{18}$$

The result we can summarize as:

$$\boxed{\begin{array}{ll} S^{\varphi}_{a} = [0\ 0\ \ldots\ 0] & S^{\varphi}_{d} = [0\ 0\ \ldots\ 0] \\ S^{\varphi}_{\alpha} = [\mathbf{x}_0\ \mathbf{x}_1\ \ldots\ \mathbf{x}_{n-1}] & S^{\varphi}_{\theta} = [\mathbf{z}_1\ \mathbf{z}_2\ \ldots\ \mathbf{z}_n] \end{array}} \tag{19}$$

By comparing (19) with (9) we notice $\mathbf{S}^{\varphi}_{\alpha} = \mathbf{S}^{p}_{a}$ and $\mathbf{S}^{\varphi}_{\theta} = \mathbf{S}^{p}_{d}$. We also notice that sensitivity matrices $\mathbf{S}^{p}_{\theta}$ and $\mathbf{S}^{\varphi}_{\theta}$ constitute the manipulator Jacobian $\mathbf{J}_{\theta}$ in the form which is originally given by Whitney [14].

4. Parallel joint axes

As pointed out by Hayati [6-8], if two consecutive joint axes are nominally parallel, small axis missalignment can cause large variations in link parameters. This invalidates standard calibration algorithms based on Denavit-Hartenberg's four-parameter model for forward kinematics. Therefore he has proposed a fifth parameter, β, which is an additional rotation of the link about its y-axis.

In order to study sensitivities respect to the new parameter, we assume the general case in which all links are described by five parameters. In this case, expressions (A-3) expand to (A-11), which gives:

$$\frac{\partial}{\partial \beta_{i-1}}\,{}^{i-1}_{i}\mathbf{R} = \mathbf{rot}(\mathbf{e}_1,\alpha_{i-1})\ \Lambda(\mathbf{e}_2)\,\mathbf{rot}(\mathbf{e}_2,\beta_{i-1})\ \mathbf{rot}(\mathbf{e}_3,\theta_i) =$$

$$= \Lambda(\mathbf{rot}(\mathbf{e}_1,\alpha_{i-1})\,\mathbf{e}_2)\ {}^{i-1}_{i}\mathbf{R} = \Lambda(\cos(\alpha_{i-1})\,\mathbf{e}_2 + \sin(\alpha_{i-1})\,\mathbf{e}_3)\ {}^{i-1}_{i}\mathbf{R}$$

Consequently:

$$\frac{\partial \mathbf{x}_i}{\partial \beta_j} = \frac{\partial}{\partial \beta_j}\left({}^{o}_{j}\mathbf{R}\ {}^{j}_{j+1}\mathbf{R}\ {}^{j+1}_{i}\mathbf{R}\,\mathbf{e}_1\right) = {}^{o}_{j}\mathbf{R}\,\Lambda(\cos(\alpha_j)\,\mathbf{e}_2 + \sin(\alpha_j)\,\mathbf{e}_3)\ {}^{j}_{i}\mathbf{R}\,\mathbf{e}_1 =$$

$$= \Lambda(\cos(\alpha_j)\,\mathbf{y}_j + \sin(\alpha_j)\,\mathbf{z}_j)\,\mathbf{x}_i. \tag{20}$$

Comparing (20) with (A-12) we see that the argument of the Λ-operator is $\mathbf{v}_j$. Hence:

$$\frac{\partial \mathbf{x}_i}{\partial \beta_j} = \begin{cases} -\mathbf{x}_i \times \mathbf{v}_j & j<i \\ 0 & j \geq i \end{cases} .$$

Similarly we get:

$$\frac{\partial \mathbf{z}_i}{\partial \beta_j} = \begin{cases} -\mathbf{z}_i \times \mathbf{v}_j & j<i \\ 0 & j \geq i \end{cases} .$$

These partial derivatives we use to obtain the positional sensitivity vector with respect to β_j:

$$\frac{\partial \mathbf{p}}{\partial \beta_j} = \mathbf{v}_j \times \rho_j , \quad \rho_j = \mathbf{r}_j - a_j \mathbf{x}_j = \mathbf{r}_{j+1} + d_{j+1} \mathbf{z}_{j+1} .$$

Using the method shown in the preceding section, we also find the orientation sensitivity vector with respect to β_j:

$$\frac{\partial \varphi}{\partial \beta_j} = \mathbf{v}_j .$$

Since the five-parameter model is used with nominally parallel joints, that is $\alpha_j = 0$, $\mathbf{v}_j$ will be identical to $\mathbf{y}_j$ (see (A-12)). This finally gives:

$$\boxed{\mathbf{S}^{p}_{\beta} = [\, \mathbf{y}_0 \times \rho_0 \;\; \mathbf{y}_1 \times \rho_1 \; \ldots \; \mathbf{y}_{n-1} \times \rho_{n-1} \,] \qquad \mathbf{S}^{\varphi}_{\beta} = [\, \mathbf{y}_0 \;\; \mathbf{y}_1 \; \ldots \; \mathbf{y}_{n-1} \,]}$$

5. Second-order sensitivity vectors

The second-order sensitivity vectors we define as partial derivatives of the first order sensitivity vectors obtained in third section. From (8) or (9) it is clear that:

$$\frac{\partial^2 \mathbf{p}}{\partial a_i \partial a_j} = \frac{\partial^2 \mathbf{p}}{\partial a_i \partial d_j} = \frac{\partial^2 \mathbf{p}}{\partial d_i \partial a_j} = \frac{\partial^2 \mathbf{p}}{\partial d_i \partial d_j} = 0.$$

In addition, from (14) and (15) we get:

$$\frac{\partial^2 \mathbf{p}}{\partial \alpha_i \partial a_j} = \begin{cases} \mathbf{x}_i \times \mathbf{x}_j & i<j \\ 0 & i \geq j \end{cases} , \qquad \frac{\partial^2 \mathbf{p}}{\partial \alpha_i \partial d_j} = \begin{cases} \mathbf{x}_i \times \mathbf{z}_j & i<j \\ 0 & i \geq j \end{cases} ,$$

$$\frac{\partial^2 \mathbf{p}}{\partial \theta_i \partial a_j} = \begin{cases} \mathbf{z}_i \times \mathbf{x}_j & i \leq j \\ 0 & i > j \end{cases} , \qquad \frac{\partial^2 \mathbf{p}}{\partial \theta_i \partial d_j} = \begin{cases} \mathbf{z}_i \times \mathbf{z}_j & i<j \\ 0 & i \geq j \end{cases} .$$

In order to obtain the other second-order sensitivities, we find first the derivatives of $\mathbf{r}_j$. Starting from (A-9), and knowing that $\mathbf{x}_i$ and $\mathbf{z}_i$ are independent from the link distances and link offsets, we can directly write:

$$\frac{\partial \mathbf{r}_j}{\partial a_i} = \begin{cases} 0 & i < j \\ \mathbf{x}_i & i \geq j \end{cases} , \qquad \frac{\partial \mathbf{r}_j}{\partial d_i} = \begin{cases} 0 & i \leq j \\ \mathbf{z}_i & i > j \end{cases} .$$

For the link twists we have:

$$\frac{\partial \mathbf{r}_j}{\partial \alpha_i} = \sum_{k=j+1}^{n} \left(\frac{\partial \mathbf{x}_{k-1}}{\partial \alpha_i} \mathrm{a}_{k-1} + \frac{\partial \mathbf{z}_k}{\partial \alpha_i} \mathrm{d}_k \right) . \tag{21}$$

Applying (11) and (12), and assuming $i < j$, (21) becomes:

$$\frac{\partial \mathbf{r}_j}{\partial \alpha_i} = \sum_{k=j+1}^{n} [(\mathbf{x}_i \times \mathbf{x}_{k-1})\, \mathrm{a}_{k-1} + (\mathbf{x}_i \times \mathbf{z}_k)\, \mathrm{d}_k] = \mathbf{x}_i \times \sum_{k=j+1}^{n} (\mathbf{x}_{k-1}\, \mathrm{a}_{k-1} + \mathbf{z}_k\, \mathrm{d}_k) = -\mathbf{r}_j \times \mathbf{x}_i, \tag{22}$$

For $i = j$ we write (see (A-9)):

$$\frac{\partial \mathbf{r}_j}{\partial \alpha_i} = \frac{\partial}{\partial \alpha_j}\Big(\mathbf{x}_j \mathrm{a}_j + \mathbf{z}_{j+1}\, \mathrm{d}_{j+1} + \mathbf{r}_{j+1} \Big) = \frac{\partial \mathbf{x}_j}{\partial \alpha_j} \mathrm{a}_j + \frac{\partial \mathbf{z}_{j+1}}{\partial \alpha_j} \mathrm{d}_{j+1} + \frac{\partial \mathbf{r}_{j+1}}{\partial \alpha_j}$$

Applying (11), (12) and (22), we get:

$$\frac{\partial \mathbf{r}_j}{\partial \alpha_i} = -(\mathbf{z}_{j+1}\, \mathrm{d}_{j+1} + \mathbf{r}_{j+1}) \times \mathbf{x}_j.$$

Since $\mathbf{x}_j \times \mathbf{x}_j = 0$ we have:

$$\frac{\partial \mathbf{r}_j}{\partial \alpha_i} = -(\mathbf{z}_{j+1}\, \mathrm{d}_{j+1} + \mathbf{x}_j \mathrm{a}_j + \mathbf{r}_{j+1}) \times \mathbf{x}_j = -\mathbf{r}_j \times \mathbf{x}_j, \qquad i = j. \tag{23}$$

Combining (22) and (23) we finally obtain:

$$\frac{\partial \mathbf{r}_j}{\partial \alpha_i} = \begin{cases} -\mathbf{r}_j \times \mathbf{x}_i & i \le j \\ -\mathbf{r}_i \times \mathbf{x}_i & i > j \end{cases} . \tag{24}$$

Similarly we obtain:

$$\frac{\partial \mathbf{r}_j}{\partial \theta_i} = \begin{cases} -\mathbf{r}_j \times \mathbf{z}_i & i \le j \\ -\mathbf{r}_i \times \mathbf{z}_i & i > j \end{cases} .$$

Using now (16) we get:

$$\frac{\partial^2 \mathbf{p}}{\partial \alpha_i \partial \alpha_j} = \frac{\partial}{\partial \alpha_i}\left(\frac{\partial \mathbf{p}}{\partial \alpha_j}\right) = \frac{\partial}{\partial \alpha_i}(\mathbf{x}_j \times \mathbf{r}_j) = \frac{\partial \mathbf{x}_j}{\partial \alpha_i} \times \mathbf{r}_j + \mathbf{x}_j \times \frac{\partial \mathbf{r}_j}{\partial \alpha_i} . \tag{25}$$

Assuming $i < j$ and applying (11) and (24), (25) becomes:

$$\frac{\partial^2 \mathbf{p}}{\partial \alpha_i \partial \alpha_j} = -(\mathbf{x}_j \times \mathbf{x}_i) \times \mathbf{r}_j - \mathbf{x}_j \times (\mathbf{r}_j \times \mathbf{x}_i). \tag{26}$$

Since $(\mathbf{a} \times \mathbf{b}) \times \mathbf{c} = \mathbf{a} \times (\mathbf{b} \times \mathbf{c}) - \mathbf{b} \times (\mathbf{a} \times \mathbf{c})$ for any three vectors **a**, **b** and **c** (see (A-6)), (26) becomes:

$$\frac{\partial^2 \mathbf{p}}{\partial \alpha_i \partial \alpha_j} = \mathbf{x}_i \times (\mathbf{x}_j \times \mathbf{r}_j),$$

For $i \geq j$ (25) yields:

$$\frac{\partial^2 \mathbf{p}}{\partial \alpha_i \partial \alpha_j} = \mathbf{x}_j \times (\mathbf{x}_i \times \mathbf{r}_i).$$

Thus:

$$\frac{\partial^2 \mathbf{p}}{\partial\alpha_i\,\partial\alpha_j} = \begin{cases} \mathbf{x}_i \times (\mathbf{x}_j \times \mathbf{r}_j) & i<j \\ -(\mathbf{x}_i \times \mathbf{r}_i) \times \mathbf{x}_j & i \geq j \end{cases} .$$

Similarly we can obtain other second-order derivatives:

$$\frac{\partial^2 \mathbf{p}}{\partial\theta_i\,\partial\alpha_j} = \begin{cases} \mathbf{z}_i \times (\mathbf{x}_j \times \mathbf{r}_j) & i<j \\ -(\mathbf{z}_i \times \mathbf{r}_i) \times \mathbf{x}_j & i \geq j \end{cases} \qquad \frac{\partial^2 \mathbf{p}}{\partial\theta_i\,\partial\theta_j} = \begin{cases} \mathbf{z}_i \times (\mathbf{z}_j \times \mathbf{r}_j) & i<j \\ -(\mathbf{z}_i \times \mathbf{r}_i) \times \mathbf{z}_j & i \geq j \end{cases}$$

Derivatives with reversed order of differentiation can be obtained using Schwarz's theorem for mixed derivatives. For example:

$$\left.\frac{\partial^2 \mathbf{p}}{\partial\alpha_i\,\partial\theta_j}\right|_{i<j} = \left.\frac{\partial^2 \mathbf{p}}{\partial\theta_j\,\partial\alpha_i}\right|_{j>i} = -(\mathbf{z}_j \times \mathbf{r}_j) \times \mathbf{x}_i = \mathbf{x}_i \times (\mathbf{z}_j \times \mathbf{r}_j)$$

The results for second-order positional sensitivity vectors are summarized in Table 1. The second-order orientation sensitivity vectors can be obtained by differentiating corresponding first order sensitivity vectors, which are given in (19). The results are summarized in Table 2.

It is interesting to note that second order sensitivity vectors can be expressed in terms of first-order sensitivity vectors. Comparing the results from the tables we can, for example, write:

$$\frac{\partial^2 \mathbf{p}}{\partial\alpha_i\,\partial\theta_j} = \begin{cases} \dfrac{\partial\boldsymbol{\varphi}}{\partial\alpha_i} \times \dfrac{\partial\mathbf{p}}{\partial\theta_j} & i<j \\ -\dfrac{\partial\mathbf{p}}{\partial\alpha_i} \times \dfrac{\partial\boldsymbol{\varphi}}{\partial\theta_j} & i \geq j \end{cases} .$$

This observation can be sumarized as follows:

$$\frac{\partial^2 \mathbf{p}}{\partial c_i\,\partial\xi_j} = \begin{cases} 0 & i<j \\ -\dfrac{\partial\mathbf{p}}{\partial c_i} \times \dfrac{\partial\boldsymbol{\varphi}}{\partial\xi_j} & i \geq j \end{cases} , \qquad \frac{\partial^2 \mathbf{p}}{\partial\xi_i\,\partial\eta_j} = \begin{cases} \dfrac{\partial\boldsymbol{\varphi}}{\partial\xi_i} \times \dfrac{\partial\mathbf{p}}{\partial\eta_j} & i<j \\ -\dfrac{\partial\boldsymbol{\varphi}}{\partial\eta_i} \times \dfrac{\partial\mathbf{p}}{\partial\xi_j} & i \geq j \end{cases} , \qquad \frac{\partial^2 \boldsymbol{\varphi}}{\partial\xi_i\,\partial\eta_j} = \begin{cases} \dfrac{\partial\boldsymbol{\varphi}}{\partial\xi_i} \times \dfrac{\partial\boldsymbol{\varphi}}{\partial\eta_j} & i<j \\ 0 & i \geq j \end{cases}$$

where c can be symbolically replaced by a or d, while ξ and η can be replaced by α or θ.

Table 1. SECOND-ORDER POSITIONAL SENSITIVITY VECTORS

	a_j	d_j	α_j	θ_j
a_i	0 0	0 0	0 $-\mathbf{x}_i\times\mathbf{x}_j$	0 $-\mathbf{x}_i\times\mathbf{z}_j$
d_i	0 0	0 0	0 $-\mathbf{z}_i\times\mathbf{x}_j$	0 $-\mathbf{z}_i\times\mathbf{z}_j$
α_i	$\mathbf{x}_i\times\mathbf{x}_j$ 0	$\mathbf{x}_i\times\mathbf{z}_j$ 0	$\mathbf{x}_i\times(\mathbf{x}_j\times\mathbf{r}_j)$ $-(\mathbf{x}_i\times\mathbf{r}_i)\times\mathbf{x}_j$	$\mathbf{x}_i\times(\mathbf{z}_j\times\mathbf{r}_j)$ $-(\mathbf{x}_i\times\mathbf{r}_i)\times\mathbf{z}_j$
θ_i	$\mathbf{z}_i\times\mathbf{x}_j$ 0	$\mathbf{z}_i\times\mathbf{z}_j$ 0	$\mathbf{z}_i\times(\mathbf{x}_j\times\mathbf{r}_j)$ $-(\mathbf{z}_i\times\mathbf{r}_i)\times\mathbf{x}_j$	$\mathbf{z}_i\times(\mathbf{z}_j\times\mathbf{r}_j)$ $-(\mathbf{z}_i\times\mathbf{r}_i)\times\mathbf{z}_j$

(Upper value is for $i < j$, lower value is for $i \geq j$)

Table 2. SECOND-ORDER ORIENTATION SENSITIVITY VECTORS

	a_j	d_j	α_j	θ_j
a_i	0 0	0 0	0 0	0 0
d_i	0 0	0 0	0 0	0 0
α_i	0 0	0 0	$\mathbf{x}_i\times\mathbf{x}_j$ 0	$\mathbf{x}_i\times\mathbf{z}_j$ 0
θ_i	0 0	0 0	$\mathbf{z}_i\times\mathbf{x}_j$ 0	$\mathbf{z}_i\times\mathbf{z}_j$ 0

(Upper value is for $i < j$, lower value is for $i \geq j$)

6. Conclusion

Kinematic sensitivity vectors and matrices with respect to link parameters have been defined and derived for open-loop, n DOF manipulators. Sensitivity vectors are expressed in terms of coordinate axes of manipulator links. A recursive algorithm for efficient computation of coordinate axes has been also presented. Second-order sensitivity vectors are also derived. It is shown that the second-order sensitivity vectors can be expressed as vector products of the first-order sensitivity vectors. The results obtained can be used for numeric and symbolic computation of kinematic sensitivities for a particular manipulator type.

Acknowledgement

Author would like to thank Dr. David Carlson from Department of Mathematical Sciences, San Diego State University, for helpful discussions during the final preparation of this paper. Verification of the results and many related computations were done by the help of Robot Shell.

References

[1] Asada, H and K. Hara, "Load Sensitivity Analysis and Adaptive Control of a Direct-Drive Arm", Proc. of the American Control Conference, pp. 799-804, Seattle, WA, 1986.

[2] Craig, J.J., *Introduction to Robotics: Mechanics and Control*, Addison-Wesley Publishing Co., Reading, MA, 1986.

[3] Cruz, J.B. (ed.), *System Sensitivity Analysis*, Benchmark Papers in Electrical Engineering and Computer Science, Dowden Hutchinson and Ross, Inc., Stroudsburg, PA, 1973.

[4] Denavit, J. and R.S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices:, *J. Appl. Mech.*, pp. 215-221, June 1955.

[5] Frank, P.M., *Introduction to System Sensitivity Theory* , Academic press, New York, NY, 1987.

[6] Hayati, S.A., "Robot Arm Geometric Link Parameter Estimation", Proc. of the 22nd Conferenceon Decision and Control, Vol. 3, pp. 1477-1483, 1983.

[7] Hayati, S.A. and M. Mirmirani, "Improving the Absolute Positioning Accuracy of Robot Manipulators", *Journal of Robotic Systems*, Vol. 2, No. 4, pp. 397-423, 1985.

[8] Hayati, S.A., S. Tso Kam and G. Roston, "Robot Geometri Calibration", IEEE International Conference on Robotics and Automation, Vol. 2, pp. 947-951, Philadelphia, PA, 1988.

[9] Kokotovic, P.V. and R.S. Rutman, "Sensitivity of Automatic Control Systems " (Survey), *Autom. Remote Control* (USSR), Vol. 26, pp. 727-749, 1965.

[10] Medvedev, V.S., "Mathematical Models of Robot Dynamics" , pp. 23-61, in: E.P. Popov (ed.), *Modern Robot Engineering* , MIR Publishers, Moscow, 1982.

[11] Paul, R.P., *Robot Manipulators: Mathematics, Programming, and Control*, The MIT Press, Cambridge, MA, 1981.

[12] Roth, Z.S., B.W. Mooring and B. Ravani, "An Overview of Robot Calibration", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 5, pp. 377-385, October 1987.

[13] Togai, M., "Manipulability and Sensitivity for Design and Evaluation of Industrial Robots: Kinematic Consideration", 15th ISIR , 1987.

[14] Whitney, D.E., "The mathematics of Coordinated Control of Prosthetic Arms and Manipulators", *Trans ASME , J. Dynamic Systems, Measurement and Control*, Vol. 122, pp. 303-309, 1972.

[15] Yoshikawa, T., "Manipulability of Robotic Mechanisms", Proc. of the 2nd Int. Symp. of Robotics Research, pp. 91-98, Kyoto, August 1984.

Appendix

The orientation $\mathbf{R}$ and position $\mathbf{p}$ of an n-DOF manipulator are given by:

$$\mathbf{R} = {}^{o}_{n}\mathbf{R} = \prod_{i=1}^{n} {}^{i-1}_{\;i}\mathbf{R}\,, \qquad \mathbf{p} = {}^{o}\mathbf{p}_{n} = \sum_{i=1}^{n} {}^{o}_{i-1}\mathbf{R}\,{}^{i-1}\mathbf{p}_{i}, \tag{A-1}$$

where ${}^{i-1}_{\;i}\mathbf{R}$ and ${}^{i-1}\mathbf{p}_i$ are the relative orientation matrix and position vector of the i-th link, and

$${}^{o}_{i}\mathbf{R} = {}^{o}_{i-1}\mathbf{R}\;{}^{i-1}_{\;i}\mathbf{R}\,. \tag{A-2}$$

If we suppose the four-parameter model originally proposed by Denavit and Hartenberg [4] and modified by Craig [2], then:

$${}^{i-1}_{\;i}\mathbf{R} = \mathbf{rot}(\mathbf{e}_1,\alpha_{i-1})\,\mathbf{rot}(\mathbf{e}_3,\theta_i), \qquad {}^{i-1}\mathbf{p}_i = \mathbf{e}_1 a_{i-1} + \mathbf{rot}(\mathbf{e}_1,\alpha_{i-1})\,\mathbf{e}_3\, d_i\,, \tag{A-3}$$

where $\mathbf{e}_1 = [1\ 0\ 0]^T$, $\mathbf{e}_2 = [0\ 1\ 0]^T$ and $\mathbf{e}_3 = [0\ 0\ 1]^T$.

The rotation operator used in (A-3) can be expressed in general form as rotation by angle ϕ about unit vector $\mathbf{k} = [\mathbf{k}_1\ \mathbf{k}_2\ \mathbf{k}_3\ ]^T$, $|\mathbf{k}| = 1$:

$$\mathbf{rot}(\mathbf{k},\phi) = \mathrm{I} + \sin\phi\ \Lambda(\mathbf{k}) + (1\text{-}\cos\phi)\ \Lambda(\mathbf{k})^2, \tag{A-4}$$

where $\mathbf{I}$ is the 3×3 identity matrix and $\Lambda(\mathbf{k})$ is the skew-symmetric operator:

$$\Lambda(\mathbf{k}) = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix}. \tag{A-5}$$

Note that operator Λ has the following interesting properties[10]:

$$\begin{array}{lll} \Lambda(\mathbf{a})^T = -\Lambda(\mathbf{a}) & \Lambda(\mathbf{a}+\mathbf{b}) = \Lambda(\mathbf{a}) + \Lambda(\mathbf{b}) & \Lambda(\mathbf{k})^2 = \mathbf{k}\mathbf{k}^T - \mathbf{I} \\ \Lambda(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b} & \Lambda(\mathbf{a})\ \Lambda(\mathbf{b}) - \Lambda(\mathbf{b})\ \Lambda(\mathbf{a}) = \Lambda(\Lambda(\mathbf{a})\mathbf{b}) & \Lambda(\mathbf{k})^3 = -\ \Lambda(\mathbf{k}), \\ \Lambda(\mathrm{c}\mathbf{a}) = \mathrm{c}\Lambda(\mathbf{a}) & \mathbf{B}\Lambda(\mathbf{a}) = \Lambda(\mathbf{B}\mathbf{a})\mathbf{B} & \end{array} \tag{A-6}$$

where $\mathbf{a}$ and $\mathbf{b}$ are arbitrary vectors, $\mathbf{k}$ is a unit vector, c is scalar and $\mathbf{B}$ is an orthogonal 3×3 matrix.
Using these properties, it can be shown:

$$\frac{\partial}{\partial\phi}\mathbf{rot}(\mathbf{k},\phi) = \mathbf{rot}(\mathbf{k},\phi)\ \Lambda(\mathbf{k}) = \Lambda(\mathbf{k})\ \mathbf{rot}(\mathbf{k},\phi). \tag{A-7}$$

If we denote the coordinate axes of the i-th link by: $\mathbf{x}_i = {}^0\mathbf{x}_i$, $\mathbf{y}_i = {}^0\mathbf{y}_i$ and $\mathbf{z}_i = {}^0\mathbf{z}_i$, i.e ${}^0_i\mathbf{R} = [\mathbf{x}_i\ \ \mathbf{y}_i\ \ \mathbf{z}_i]$,

$\mathbf{x}_i = {}^0_i\mathbf{R}\ \mathbf{e}_1$, $\mathbf{y}_i = {}^0_i\mathbf{R}\ \mathbf{e}_2$, $\mathbf{z}_i = {}^0_i\mathbf{R}\ \mathbf{e}_3$, then we can express $\mathbf{p}$ in terms of these vectors:

$$\mathbf{p} = \sum_{i=1}^{n}\ \mathbf{x}_{i-1}\ a_{i-1} + \mathbf{z}_i\ d_i\ . \tag{A-8}$$

Similarly we can express the distance of the i-th link from the last, n-th, link:

$$\mathbf{r}_i = \mathbf{r}_{i+1} + \mathbf{x}_i\, a_i + \mathbf{z}_{i+1}\, d_{i+1} = \sum_{j=i+1}^{n} \mathbf{x}_{j-1}\, a_{j-1} + \mathbf{z}_j\, d_j,\quad i = n\text{-}1, n\text{-}2,\ldots,1,\qquad \mathbf{r}_n = 0,\qquad \mathbf{p} = \mathbf{r}_1. \tag{A-9}$$

Note that we have assumed here coordinate assignments as proposed by Craig [2], where the z-axis of the i-th frame, $\mathbf{z}_i$, is colinear with the i-th joint axis, and the origin of the i-th link is lying on the axis.

Vectors $\mathbf{x}_i$, $\mathbf{y}_i$ and $\mathbf{z}_i$ can be computed recursively. Using (A-2), (A-3) and (A-4) we can obtain:

$$\begin{array}{ll} \mathbf{v}_i = \cos(\alpha_{i-1})\ \mathbf{y}_{i-1} + \sin(\alpha_{i-1})\ \mathbf{z}_{i-1}, & \mathbf{y}_i = c_i\mathbf{v}_i - s_i\mathbf{x}_{i-1}, \\ \mathbf{x}_i = s_i\mathbf{v}_i + c_i\mathbf{x}_{i-1}, & \mathbf{z}_i = \cos(\alpha_{i-1})\ \mathbf{z}_{i-1} - \sin(\alpha_{i-1})\ \mathbf{y}_{i-1},\qquad i = 1,\ldots,n \end{array} \tag{A-10}$$

In the five-parameter model proposed by Hayati [6-8], relative orientations and positions of links become:

$${}^{i-1}_{\ i}\mathbf{R} = \mathbf{rot}(\mathbf{e}_1,\alpha_{i-1})\ \mathbf{rot}(\mathbf{e}_2,\beta_{i-1})\ \mathbf{rot}(\mathbf{e}_3,\theta_i),\qquad {}^{i-1}\mathbf{p}_i = \mathbf{e}_1\, a_{i-1} + \mathbf{rot}(\mathbf{e}_1,\alpha_{i-1})\ \mathbf{rot}(\mathbf{e}_2,\beta_{i-1})\ \mathbf{e}_3\, d_i. \tag{A-11}$$

This will result in a similar set of recursive relations for coordinate axes:

$$\begin{array}{ll} \mathbf{v}_i = \sin(\alpha_{i-1})\ \mathbf{z}_{i-1} + \cos(\alpha_{i-1})\ \mathbf{y}_{i-1}, & \mathbf{x}_i = s_i\mathbf{v}_i + c_i\mathbf{u}_i, \\ \mathbf{w}_i = \cos(\alpha_{i-1})\ \mathbf{z}_{i-1} - \sin(\alpha_{i-1})\ \mathbf{y}_{i-1}, & \mathbf{y}_i = c_i\mathbf{v}_i - s_i\mathbf{u}_i, \\ \mathbf{u}_i = \cos(\beta_{i-1})\ \mathbf{x}_{i-1} - \sin(\beta_{i-1})\ \mathbf{w}_i, & \mathbf{z}_i = \sin(\beta_{i-1})\ \mathbf{x}_{i-1} + \cos(\beta_{i-1})\ \mathbf{w}_i,\quad i = 1,\ldots,n. \end{array} \tag{A-12}$$

Note, for $\beta_{i-1} = 0$, $\mathbf{u}_i = \mathbf{x}_{i-1}$, and (A-12) reduces to (A-10).