Two New Algorithms for Forward and Inverse Kinematics under Degenerate Conditions

Ezio MALIS **Lionel MORIN** **Sylvie BOUDET**

Direction des Etudes et Recherches d'Electricité De France
6, Quai Watier - 78401 Chatou - FRANCE.

ABSTRACT

We propose a fast algorithm for the computation of any order derivatives of robots' forward kinematics. We calculated second derivatives to show its application in real-time robotics. Then, an algorithm for solving the inverse kinematics problem at singularities under degenerate conditions is proposed. The algorithm was successfully tested on a 7 degree of freedom robot.

KEYWORDS: Kinematics, Derivatives, Degenerate Conditions, Serial Manipulators, Real-time Control.

INTRODUCTION

To solve the inverse kinematics problem we need to find the joint displacement that moves the end-effector of the robot to a specified position. The closed form solution can be obtained only for non-redundant robots with special geometry. For a general serial manipulator different iterative methods are used. The majority of the methods in the literature use the information provided by the Jacobian matrix: the Jacobian transpose method [1], the Jacobian pseudo-inverse method [2], the damped least-squares Jacobian inverse method [3], [4]. Real-time control of robots is done without difficulty with one of these methods to find joint displacements from Cartesian commands as long as the Jacobian is not singular. Suppose the robot is at a singularity and the desired motion is in a degenerate direction. In this case the robot may get stuck. This condition does not mean that the displacement along the degenerate direction cannot be performed. If the movement is possible and all the points along this direction are in the workspace, there is at least one joint position for each of these points. Nielsen [5] outlined that this solution can be found by using the second derivatives of forward kinematics (i.e. Hessian matrices) the Jacobian matrix being only the first term in a Taylor expansion of the forward kinematics of a robot. If the first term is rank deficient, other terms (i.e. the higher order derivatives) give the information needed to accomplish the desired motion. In order to perform the Taylor expansion of forward kinematics, we propose in this paper an algorithm for the real-time computation of any order derivatives of forward kinematics of any serial robot (n joints, rotary or prismatic). Egeland [6] has shown that the Cartesian acceleration term provides the information required to exit the singularity. Thus, he proposed to move the robot inside the joint null space. This is easy if the dimension of the null space is 1. But in general case for a n joints robot, its dimension is n-r (where r is the rank of the Jacobian matrix). The direction of the null space in which we have to move is difficult to know, and it is possible that a null space motion doesn't take a redundant robot out of the singularity. In order to solve the inverse kinematics problem at singularities, when the desired motion is in a degenerate direction, a new algorithm for a general serial robot is proposed in the paper.

This paper is organized as follows. In the first section we introduce a notation for orientation coordinates which facilitates the mathematics of the algorithms presented here. In the second section we illustrate the algorithm to calculate any order derivatives of forward kinematics. The final result is the real-time Hessian matrices computation. In the third section we illustrate the algorithm for inverse kinematics at singularities under degenerate conditions. The experimental results are given in the fourth section.

FORWARD KINEMATICS

To represent the kinematics of a robot, we will use the Denavit-Hartemberg modelisation modified by Dombre and Khalil [7]. The transformation matrix between the end-effector frame R_n and the reference frame R_0 is ${}^0\mathbf{T}_n$. This representation is redundant because the orientation matrix has 9 parameters, when only 3 are needed. A rotation of θ around a unitary vector **u** can be described with the following four parameters:

$$\mu_1 = \cos(\theta), \quad \mu_2 = u_x \sin(\theta), \quad \mu_3 = u_y \sin(\theta), \quad \mu_4 = u_z \sin(\theta) \tag{1}$$

If we calculate these parameters in function of the elements of the matrix ${}^0\mathbf{T}_n$ we obtain:

$$\mu_1 = \frac{1}{2}\left(s_x + n_y + a_z - 1\right),\ \mu_2 = \frac{1}{2}\left(n_z - a_y\right),\ \mu_3 = \frac{1}{2}\left(a_x - s_z\right),\ \mu_4 = \frac{1}{2}\left(s_y - n_x\right) \tag{2}$$

where $\mathbf{s} = \left[\mathrm{s_x} \quad \mathrm{s_y} \quad \mathrm{s_z}\right]^{\mathrm{T}}$, $\mathbf{n} = \left[\mathrm{n_x} \quad \mathrm{n_y} \quad \mathrm{n_z}\right]^{\mathrm{T}}$ and $\mathbf{a} = \left[\mathrm{a_x} \quad \mathrm{a_y} \quad \mathrm{a_z}\right]^{\mathrm{T}}$ are the columns of the orientation matrix. Euler's parameters are not used in this algorithm because they are non-linear functions of the elements of the transformation matrix ${}^0\mathbf{T}_n$. The notation uses the advantages of the homogeneous matrix for transformation multiplication and those of linear vector expressions for deriving easily forward kinematics. Furthermore, if we make the hypothesis that $-\pi/2 \le \theta \le \pi/2$, the first parameter can be discarded and the representation of the rotation is minimal. The commanded motions on a robot in one sampling time are usually not large. If the rotation is larger than π, it can be split into two rotations. The vectorial form of the forward kinematics of a robot is:

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) = \left[p_x \quad p_y \quad p_z \quad \frac{1}{2}\left(n_z - a_y\right) \quad \frac{1}{2}\left(a_x - s_z\right) \quad \frac{1}{2}\left(s_y - n_x\right) \right]^T \tag{3}$$

this is a set of linear functions of the elements of the transformation matrix ${}^0\mathbf{T}_n$.

DIFFERENTIAL KINEMATICS

We take the p order Taylor development of the transformation matrix ${}^{0}\mathbf{T}_n$ calculated in $\mathbf{q}$. Let $\tilde{\mathbf{q}} = \mathbf{q} + \delta\mathbf{q}$ so that:

$$ {}^{0}\mathbf{T}_n(\tilde{\mathbf{q}}) = {}^{0}\mathbf{T}_n(\mathbf{q}) + \sum_{j=1}^{n} \frac{\partial\left[{}^{0}\mathbf{T}_n(\mathbf{q})\right]}{\partial q_j}\delta q_j + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2\left[{}^{0}\mathbf{T}_n(\mathbf{q})\right]}{\partial q_i \partial q_j}\delta q_i \delta q_j + \ldots + \mathbf{R}_p(\mathbf{q}) \qquad (4) $$

To know the general derivative of the matrix ${}^{0}\mathbf{T}_n$ we use the following theorem:

Theorem: If the reference frame is the initial end-effector frame:

$$ {}^{0}\mathbf{T}_n(\tilde{\mathbf{q}})\Big|_{\tilde{\mathbf{q}}=\mathbf{q}} = {}^{0}\mathbf{T}_n(\mathbf{q}) = \mathbf{I} \qquad (5) $$

the general derivative of the forward kinematics may be calculated as follows:

$$ \left.\frac{\partial^p\left[{}^{0}\mathbf{T}_n(\tilde{\mathbf{q}})\right]}{\partial\tilde{q}_1^{p_1}\partial\tilde{q}_2^{p_2}\cdots\partial\tilde{q}_n^{p_n}}\right|_{\tilde{\mathbf{q}}=\mathbf{q}} = \left(\left.\frac{\partial\left[{}^{0}\mathbf{T}_n(\tilde{\mathbf{q}})\right]}{\partial\tilde{q}_1}\right|_{\tilde{\mathbf{q}}=\mathbf{q}}\right)^{p_1}\left(\left.\frac{\partial\left[{}^{0}\mathbf{T}_n(\tilde{\mathbf{q}})\right]}{\partial\tilde{q}_2}\right|_{\tilde{\mathbf{q}}=\mathbf{q}}\right)^{p_2}\cdots\left(\left.\frac{\partial\left[{}^{0}\mathbf{T}_n(\tilde{\mathbf{q}})\right]}{\partial\tilde{q}_n}\right|_{\tilde{\mathbf{q}}=\mathbf{q}}\right)^{p_n} \qquad (6) $$

where $p = p_1 + p_2 + \ldots + p_n$ and p_j is a positive or null integer.

This theorem show that all derivatives can be calculated only by calculating the first derivatives.

First derivatives: the Jacobian matrix

The Jacobian matrix is calculated by deriving the forward kinematics. We will note with j the j-th columns of the Jacobian matrix. The subscript "p" indicates the position and the subscript "o" the orientation. The forward kinematics defined by equation (3) is a set of linear functions of the elements of the transformation matrix ${}^{0}\mathbf{T}_n$. If the reference frame is the initial end-effector frame, the elements of the j-th column of the Jacobian matrix are obtained directly by using equation (6) for p=1. The first derivative of the matrix ${}^{0}\mathbf{T}_n$ can be written as follows:

$$ \left.\frac{\partial\left[{}^{0}\mathbf{T}_n(\tilde{\mathbf{q}})\right]}{\partial\tilde{q}_j}\right|_{\tilde{\mathbf{q}}=\mathbf{q}} = \begin{bmatrix} {}^{j}\hat{\mathbf{g}}_o & {}^{j}\mathbf{g}_p \\ \mathbf{0} & 0 \end{bmatrix} \qquad (7) $$

where ${}^{j}\hat{\mathbf{g}}_o$ is the skew matrix associated to the vector ${}^{j}\mathbf{g}_o$. The terms of the Jacobian matrix in this vector are the derivatives with respect to q_j of rotation around axis x, y and z. Equation (7) is available for both prismatic and rotary joint. The difference between rotary and prismatic joints is made when the Jacobian matrix elements are calculated.

Second derivatives: the Hessian matrices

Let $h^{k}_{i,j}$ be the (i,j) element of the k-th Hessian matrix $\mathbf{H}_k$ ($\{k = 1,2,...,6\}$):

$$h^{k}_{i,j} = \frac{\partial f_k(\mathbf{q})}{\partial q_i \partial q_j} \tag{8}$$

We calculate only elements with $1 \le i \le n$ and $i \le j \le n$ because the transformation matrix is a continuous function of $\mathbf{q}$ and so by using the Schwartz theorem, we have $h^{k}_{j,i} = h^{k}_{i,j}$. As for first derivatives, the (i,j) element of the Hessian matrix are obtained directly from equation (6) for p=2. By using equation (7), the second derivatives can be written in function of the elements of the Jacobian matrix:

$$\left.\frac{\partial^2 \left[{}^{0}\mathbf{T}_n(\tilde{\mathbf{q}})\right]}{\partial \tilde{q}_i \partial \tilde{q}_j}\right|_{\tilde{\mathbf{q}}=\mathbf{q}} = \begin{bmatrix} {}^{i}\hat{\mathbf{g}}_o & {}^{i}\mathbf{g}_p \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} {}^{j}\hat{\mathbf{g}}_o & {}^{j}\mathbf{g}_p \\ \mathbf{0} & 0 \end{bmatrix} \tag{9}$$

This equation is available for both prismatic and rotary joint. The difference between rotary and prismatic joints is made when the elements Jacobian matrix are calculated.

INVERSE KINEMATICS

The algorithm proposed in this paper minimizes the following function:

$$\min_{\delta\mathbf{q}} \quad \Phi = \left\|\delta\mathbf{q}\right\|^2 \tag{10}$$

subject to:

$$\min_{\delta\mathbf{q}} \Psi = \left\|\delta\mathbf{x} - \left(\mathbf{f}(\mathbf{q}+\delta\mathbf{q}) - \mathbf{f}(\mathbf{q})\right)\right\|^2 \tag{11}$$

Our idea is to take a second order Taylor expansion of the norm of the Cartesian error:

$$\hat{\Psi} = \delta\mathbf{x}^T \delta\mathbf{x} - 2\delta\mathbf{x}^T \mathbf{J}(\mathbf{q})\delta\mathbf{q} + \frac{1}{2}\delta\mathbf{q}^T \left[2\mathbf{J}^T(\mathbf{q})\mathbf{J}(\mathbf{q}) - \sum_{i=1}^{m} \delta x_i \mathbf{H}_i(\mathbf{q})\right]\delta\mathbf{q} \tag{12}$$

We define C, **B** and **E** in the following way:

$$C = \delta\mathbf{x}^T\delta\mathbf{x}, \quad \mathbf{B} = -2\delta\mathbf{x}^T\mathbf{J}(\mathbf{q}), \quad \mathbf{E} = 2\mathbf{J}^T(\mathbf{q})\mathbf{J}(\mathbf{q}) - \sum_{i=1}^{m}\delta x_i \mathbf{H}_i(\mathbf{q}) \tag{13}$$

Equation (12) gives a better approximation of the Cartesian error than other methods, thanks to the term using the Hessian matrices. We test the vector **B** of equation (13). If $\mathbf{B} \neq \mathbf{0}$, we can use any of the existing algorithms. If $\mathbf{B} = \mathbf{0}$ and $\delta\mathbf{x} \neq \mathbf{0}$, $\delta\mathbf{x}$ is a degenerate vector, we cannot use those algorithms, and we have to solve the set of equations (10) and (11) written hereafter. The minimum value of Ψ is 0, thus our problem is equivalent to minimize equation (10) with the following constraint:

$$\hat{\Psi} = \mathrm{C} + \frac{1}{2}\delta\mathbf{q}^{\mathrm{T}}\mathbf{E}\delta\mathbf{q} = 0 \tag{14}$$

The Lagrange operator of equations (10) and (14) is:

$$L = \delta\mathbf{q}^{T}\delta\mathbf{q} + \lambda\left(C + \frac{1}{2}\delta\mathbf{q}^{T}\mathbf{E}\delta\mathbf{q}\right) \tag{15}$$

The necessary conditions for solving this set of equations are:

$$\frac{\partial \mathrm{L}}{\partial \delta\mathbf{q}} = \delta\mathbf{q} + \lambda\mathbf{E}\delta\mathbf{q} = 0 \tag{16}$$

$$\frac{\partial \mathrm{L}}{\partial \lambda} = \mathrm{C} + \frac{1}{2}\delta\mathbf{q}^{\mathrm{T}}\mathbf{E}\delta\mathbf{q} = 0 \tag{17}$$

If $\lambda = 0$, equation (16) implies $\delta\mathbf{q} = \mathbf{0}$ and equation (17) doesn't hold when $\delta\mathbf{x} \neq \mathbf{0}$. So λ is not null and μ is defined as $\mu = -1/\lambda$. Equation (16) can be expressed as follow: $\mathbf{E}\delta\mathbf{q} = \mu\delta\mathbf{q}$. Therefore $\delta\mathbf{q}$ is an eigen vector of $\mathbf{E}$ and μ is its eigen value. We choose a normalized vector $\mathbf{v}$ and a scalar α so that $\delta\mathbf{q} = \alpha\mathbf{v}$. By plugging this equation into equation (17), we get:

$$\mathrm{C} + \frac{1}{2}\alpha^{2}\mu = 0 \tag{18}$$

C is positive, therefore we have to choose a negative eigen value of **E** to verify equation (18). The solution to equation (18) is:

$$\alpha = \pm\sqrt{-\frac{2\mathrm{C}}{\mu}} \tag{19}$$

The bigger is the norm of the eigen value μ, the smaller is α. Since **v** is normalized the norm of α is the norm of $\delta\mathbf{q}$. We want the smallest norm of $\delta\mathbf{q}$ to satisfy equation (10), so we have to choose the lowest eigen value of **E**. The sign of α determines the configuration of the robot when it exits the singularity.

EXPERIMENTAL RESULTS

The algorithm was tested on the PA-10 from Mitsubishi, a 7 degree of freedom robot. The Singular Value Decomposition (S.V.D.) algorithm was used to calculate the eigen values and vectors of **E**. The pseudo-inverse algorithm was used to inverse the Jacobian matrix.The computation period is 20 milliseconds on a MVME 167 processor card (VME bus). Suppose now the robot in a singularity. We want to move the end-effector of the robot along a degenerate direction. When we use the pseudo-inverse algorithm the robot doesn't move because we filter noise on joint measurements. If we don't filter the joint measurements, it is possible that robot moves but is not possible to control when movement occurs nor the configuration of the robot. On the other hand when we use also

Hessian information the Cartesian error norm is small and the movement start immediately. It is therefore possible to move away from a singular configuration in a degenerate direction with little tracking error. The lower the initial velocity the lower the initial Cartesian error norm. We have carried out several tests at every singularity with similar results: we can reduce the Cartesian error as we wish if we accept a lower initial velocity and we can choose the initial configuration of the robot. If there are no negative eigen values, the Taylor expansion needs to be taken to a higher order, and a new algorithm is needed. In the case of the seven degree of freedom Mitsubishi robot arm, our experimental results confirmed that there is at least one negative eigen value at each singularity.

CONCLUSIONS

The first algorithm described in this paper allows recursive computation of all derivatives of the robot kinematics from their first derivatives. In this way, the Hessian matrices are easily obtained from the Jacobian matrix. To calculate all Hessian matrices of an n joints serial robot, we need $6n^2$ multiplication and $3n^2$ additions. With this algorithm, it possible to calculate in real-time, the Hessians from closed form equations, with significant improvement of performance of robots around singularities. The second algorithm described in this paper solves the problem of moving a general manipulator in a degenerate direction from a singularity of its workspace. The basic idea is based on taking a Taylor expansion of the norm of the function to minimize. To illustrate the algorithm, a second order expansion was performed in this paper. There may be robots for which higher orders are needed. Using the first algorithm given in the paper we can calculate the necessary matrices used for solving the problem at any order.

REFERENCES

1. Chiaverini S., Sciavicco L., Siciliano B.: "Control of Robotic Systems through Singularities", *Proc. of the Int. Workshop on Nonlinear and Adaptive Contr.: Issues in Robotics*, Grenoble, France (Nov. 1990), 285-294.
2. Withney D.E.: "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators", *Trans. ASME J. Dynamic Systems, Measurement and Control*, 94 (Dec. 1972), 303-309.
3. Nakamura Y., Hanafusa H.: "Inverse kinematics solutions with singularity robustness for robot manipulator control", *Trans. ASME, J. Dynamic Syst., Meas. and Contr.*, 108 (Sept. 1986), 163-171.
4. Wampler C.W.: "Manipulator Inverse Kinematics Solution Based on Damped Least-squares Solutions", *IEEE Trans. Systems, Man and Cybernetics*, 16(1) (Jan. 1986), 93-101.
5. Nielsen L., Canudas de Wit C., Hagander P.: "Controllability Issues of Robots near Singular Configurations", *Proc. of the Int. Workshop on Nonlinear and Adaptive Control: Issues in Robotics,* Grenoble, France, (Nov. 1990), 307-314.
6. Egeland O., Spangelo I.: "Manipulator Control in Singular Configurations - Motion in Degenerate Directions", *Proc. of the Int. Workshop on Nonlinear and Adaptive Control: Issues in Robotics,* Grenoble, France, (Nov. 1990), 296-306.
7. Dombre E., Khalil W.: "Modélisation et commande des robots", *Série Robotique* Hermès, (1986).